Universidade Federal de Santa Catarina – UFSC

Curso de Pós-Graduação em Ciência da Computação

GERÊNCIA DE DESEMPENHO DO TRÁFEGO EM REDES UTILIZANDO BASELINE BAYESIANA

Cleverson Alessandro Veronez

Dissertação submetida à Universidade Federal de Santa Catarina para a obtenção do grau de Mestre em Ciência da Computação

Profª. Sílvia Modesto Nassar

Orientadora

Florianópolis, Fevereiro de 2000

GERÊNCIA DE DESEMPENHO DO TRÁFEGO EM REDES UTILIZANDO BASELINE BAYESIANA

Cleverson Alessandro Veronez

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE

MESTRE EM CIÊNCIA DA COMPUTAÇÃO

ÁREA DE CONCENTRAÇÃO SISTEMAS DE COMPUTAÇÃO E APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO – CPGCC.

Fernando Álvaro Ostuni Gauthier, Dr.
Coordenador do Curso

Banca Examinadora:

Sílvia Modesto Nassar, Drª.
Orientadora

Prof. Carlos Becker Westphall, Dr.
Coorientador

Prof. Jorge Muniz Barreto, Dr.

Prof. João Bosco Mangueira Sobral, Dr.

Com muito amor e carinho a meus queridos pais Adhemar Veronez e Maria Ecinete Leite Veronez, que deram tudo de si, para me incentivar e orientar com um espírito de superação contínua na vida.

À minha irmã Keila Naiara Veronez que sempre me apoiou nas horas difíceis.

AGRADECIMENTOS

À Deus, pois quando tudo parece estar perdido ainda podemos encontrar forças na fé.

Aos meus pais e irmã que têm me apoiado em todos os meus empreendimentos.

À minha querida orientadora Sílvia e ao meu coorientador Westphall, mais que orientadores, amigos em todas as horas, sem vocês este trabalho não seria possível. Em todo momento me deram a certeza que eu tinha um alicerce forte e que eu poderia contar sempre com vocês.

Ao prof. Barreto e ao prof. Bosco, que em festas foram amigos e na defesa fizeram parte da banca.

Aos meus professores, desde o jardim de infância até hoje.

Aos meus amigos da INF, aos quais gostaria de poder agradecer individualmente.

Ao LRG, LISA, LabPos, INTEROP, laboratórios que ofereceram condições e equipamentos de trabalho.

Às funcionárias da PGCC Verinha e Val.

À Maria do cafézinho.

À UFSC - CPGCC e à CAPES, que proporcionaram as condições necessárias para a realização deste trabalho.

Ao Edson Mello que contribuiu muito como banca do meu TI, orientou-me e aceitou-me como estagiário do INTEROP.

A todos os funcionários e estagiários do NPD que colaboraram para meu aprendizado prático em redes e gerência.

À Marlene e ao Renato, amigos que zelaram por minha saúde.

Aos meus mestres e amigos da capoeira, mesmo com um aluno iniciante pode-se aprender grandes lições. Agradeço especialmente aos amigos do Grupo de Shows, que fizeram dos ensaios e apresentações momentos de diversão e prazer.

A todos que moraram comigo e aos ótimos momentos.

E a todas as pessoas que de alguma forma passaram por minha vida e "fizeram a diferença" contribuindo para moldar o meu caráter, contribuindo de alguma forma com a realização deste trabalho.

RESUMO

Mesmo utilizando as facilidades das plataformas de gerência convencionais, como monitoração de variáveis e visualizações gráficas, gerenciar uma rede de computadores qualquer ainda é uma tarefa difícil. A todo momento o administrador, que tem suas decisões apoiadas basicamente em dados, depara-se com situações que exigem tratamento da incerteza. Apesar dos serviços oferecidos, a maioria das plataformas atuais não são capazes de identificar com precisão o problema e de sugerir ações corretivas, deixando ao administrador o encargo de interpretar gráficos e valores de variáveis. Neste contexto, este trabalho propõe um estudo da aplicação do raciocínio probabilístico, através da implementação de uma rede bayesiana de conhecimento, a qual é capaz de reconhecer os relacionamentos entre as variáveis que representam o tráfego da rede. O resultado deste trabalho tem como objetivo oferecer suporte ao administrador no processo de tomada de decisão.

ABSTRACT

Although using the means of conventional management platforms for monitoring variables and plotting graphs, the management of a network remains a difficult task. At every moment the administrator, which has his/her decisions supported basically on data, comes across situations that demand treatment of uncertainty. In spite of the offered services, any current management platform is not capable of identifying problems and suggesting corrective actions, and as a consequence leaves to the administrator the burden of interpreting graphs and variable values. Within this context, this work presents a study on the application of probabilistic reasoning, through the implementation a bayesian belief network, which is capable of recognizing the relationships among the figures that represent traffic of the network. In addition, the knowledge bayesian network may also be used as a dynamic baseline for proactive expert systems. The outcome of this research resulted in a mechanism, which aims at supporting the administrator in the decision-making process.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACAFE – Associação Catarinense das Fundações Educacionais

ASN.1 – *Abstract Systax Notation*.1

bps – bits por segundo

CBR – *Case Based Reasoning*

CCITT – *Comité Consultatif International Télégraphique et Téléphonique*

CMIP – *Commom Management Information Protocol*

CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico

CxBus – *Cisco Extended Bus*

DM – *Data Mining*

DOD – Departamento de Defesa dos Estados Unidos

EGP – *Exterior Gateway Protocol*

EPAGRI – Empresa de Pesquisa Agropecuária e Difusão de Tecnologia de Santa Catarina

FIESC – Federação das Industrias de Santa Catarina

FMU – *Functional Management Unit*

HP – *Hewlett Packard*

IAB – *Internet Activities Board*

ICMP – *Internet Control Message Protocol*

IEEE – *Institute of Electical and Electronics Engineers*

IP – *Internet Protocol*

ISO – *International Organization for Standardization*

KDD – *Knowledge Data Discovery*

MCT – Ministério da Ciência e Tecnologia

MIB – *Management Information Base*

MU - *Management Unit*

OID – *Object Identifier*

OLAP – *On-Line Analytical Processing*

OSI – *Open Systems Interconection*

PC – *Personal Computer*

PDU – *Protocol Data Unit*

RCT – Rede Catarinense de Ciência e Tecnologia

RFC – *Request for Comments*

RNP – Rede Nacional de Pesquisa

SC – Santa Catarina

SEBRAE – Serviço de Apoio à Pequena e Média Empresa

SED – Secretaria de Educação, Cultura e Desporto

SETEMA – Secretaria de Estado da Tecnologia, Energia e Meio-Ambiente

SISGEBAY - Sistema de Gerência de Redes Bayesiano

SMFA – *Specific Management Functional Area*

SNMP – *Simple Network Management Protocol*

TCP – *Transmission Control Protocol*

UDESC – Universidade do Estado de Santa Catarina

UDP – *User Datagram Protocol*

UFSC – Universidade Federal de Santa Catarina

CAPÍTULO I

1. Introdução

1.1. Apresentação

Uma rede pode existir sem mecanismos de gerenciamento, todavia seu uso pode encontrar dificuldades com congestionamento, segurança, roteamento, etc. [ROCHA 97]. O gerenciamento é usado para controlar as atividades e monitorar os recursos da rede. Simplificando, o trabalho básico da gerência de rede é obter informação, extraída de dados, para um possível diagnóstico e execução de ações para resolver os problemas. Para alcançar estes objetivos, as funções de gerenciamento devem estar contidas em diversos componentes da rede permitindo o diagnóstico, a prevenção e a reação para problemas [WESTPHALL 91,WESTPHALL 96].

No trabalho de gerenciar uma rede de computadores existe incerteza e o uso da inteligência artificial pode ser justificado pelas vantagens que adicionam a um sistema de gerência de redes, tais como:

- A tarefa do administrador é facilitada, provendo uma melhor desempenho, pois o sistema especialista pode alcançar todos os segmentos da rede;
- O sistema se torna mais ágil, com baixo custo e grande produtividade na execução dos serviços monitorados;

- O tempo de tomada de decisão é reduzido, uma vez que o sistema notifica o gerente e propõe possíveis ações a serem tomadas;

- O tempo necessário para treinamento de um gerente de rede é reduzido sensivelmente.

Existem basicamente quatro linhas de estudos na representação do conhecimento incerto [LAURITZEN 88]. O modelo lógico, se utiliza apenas de processamento simbólico [COHEN 85]. O modelo lingüístico baseia-se no raciocínio *fuzzy* para interpretar sentenças imprecisas da linguagem natural [ZADEH 83]. A visão legal se utiliza das funções de crença da teoria de Dempster-Shafer [SHAFER 76] e o modelo bayesiano está baseado no cálculo de probabilidades. O modelo bayesiano, por utilizar a teoria da probabilidade é consistente e confiável [LINDLEY 82] possuindo um forte apelo pragmático, já que possui flexibilidade e meios operacionais de avaliação, crítica e aprendizado de dados [CHEESEMAN 85], além de prover uma metodologia muito adequada à compreensão humana.

O método Bayesiano passou a ser aplicado em sistemas especialistas sendo uma teoria consistente e que permite a representação de conhecimentos certos e incertos. A maior dificuldade encontrada foi o grande esforço computacional exigido, pois no cálculo das distribuições de probabilidade há uma explosão combinatória. Mas, quando é explorada a esparsidade das relações entre as variáveis, este esforço computacional é reduzido [PEARL 88].

Abordando a gerência de redes pela visão estatística, o presente trabalho propõe a implementação de uma rede bayesiana, a qual é capaz de reconhecer os relacionamentos entre as variáveis que representam o tráfego da rede e é capaz de estimar o vetor de probabilidades de diferentes estados de comportamento da rede. Em acréscimo, a rede bayesiana poderá também ser utilizada como uma *baseline* dinâmica para sistemas especialistas de gerência proativos.

1.2. Objetivos

1.2.1. Objetivo Geral

Como objetivo geral deste trabalho, busca-se explorar a aplicação do raciocínio bayesiano no apoio à gerência de redes de computadores[1].

1.2.2. Objetivos Específicos

Tem-se como objetivos específicos:

- Monitorar um segmento de rede;
- Criar uma base de dados com os valores das variáveis monitoradas;
- Descobrir o relacionamento entre as variáveis monitoradas;
- Aplicar técnicas de Data Mining na base de dados para criar a *baseline* bayesiana;
- Desenvolver um sistema, utilizando a *baseline* bayesiana, capaz de predizer o comportamento do segmento monitorado da rede.

---

[1] Como poderá ser constatado no decorrer do trabalho, as variáveis monitoradas oferecem dados sobre o tráfego do segmento monitorada da rede, não importando qual protocolo de comunicação a rede faz uso. Portanto este trabalho pode ser aplicado em qualquer rede de computadores (FDDI, Ethernet, ATM, etc.) desde que possua equipamentos gerenciáveis e que aceitem o protocolo SNMP.

1.3. Estrutura do Trabalho

Os próximos capítulos deste trabalho versam sobre o Raciocínio Probabilístico, o Domínio de Aplicação e a Proposta de Dissertação.

O capítulo 2 apresenta o Estado da Arte em Sistemas Especialistas Probabilísticos e também alguns trabalhos correlatos na área de gerência de redes e inteligência artificial.

O capítulo 3 relata a fundamentação teórica desta pesquisa, dando ênfase à Probabilidade Bayesiana, *Data Mining* e Gerenciamento de Redes.

No capítulo 4, foi detalhado o domínio de aplicação desta pesquisa.

O capítulo 5 apresenta o sistema implementado, onde são mostrados o ambiente de trabalho, a arquitetura do sistema, a metodologia de desenvolvimento, os programas e os resultados práticos deste trabalho.

O capítulo 6 apresenta as conclusões e perspectivas futuras deste trabalho, seguido das referências bibliográficas utilizadas.

CAPÍTULO II

2. O Estado da Arte

Neste capítulo são discutidas algumas questões sobre a abordagem bayesiana em Sistemas Especialistas e são apresentados alguns trabalhos que também fizeram uso da inteligência artificial para a gerência de redes.

2.1. Sistemas Especialistas Bayesianos

Especialistas humanos são capazes de tomar decisões baseados em informação incerta, incompleta e, até mesmo contraditória. Para que um sistema especialista seja confiável, o mesmo deve lidar com este tipo de informação com a mesma facilidade que o ser humano.

Depois da metade da década de 80, a pesquisa sobre raciocínio probabilístico em sistemas especialistas resultou na introdução das Redes Bayesianas, também chamadas de Redes Causais. Estas redes têm sua origem na teoria da probabilidade e são caracterizadas por um poderoso formalismo que representa o conhecimento no domínio e pelas incertezas associadas a este domínio. Mais especificamente, o formalismo proporciona uma

representação concisa de uma distribuição conjunta de probabilidades em um grupo de variáveis. Associados a este formalismo estão os algoritmos para calcular eficientemente as probabilidades relevantes e para processar as evidências; estes algoritmos constituem os blocos básicos para o raciocínio com o conhecimento. Desde sua introdução, a estrutura de redes bayesianas vem ganhando popularidade e está começando a mostrar o seu valor em domínios complexos. Aplicações práticas estão sendo desenvolvidas, por exemplo, para diagnóstico e prognóstico médico; para recuperação de informação probabilística e para visão em computadores [LINDA 96].

Nos sistemas especialistas bayesianos os valores das probabilidade refletem a crença do especialista sobre o que espera que ocorra em situações similares às que tem experienciado e aprendido. A idéia de que as probabilidades se alteram com a mudança de conhecimento é crucial para estes sistemas. Eles têm em sua base de conhecimentos fatos e regras que representam o conhecimento do especialista num domínio de aplicação. Aos fatos e às regras são associadas às incertezas presentes no domínio, e é explicitada a crença em sua ocorrência através de valores de probabilidade. O raciocínio realizado pelo sistema deve considerar estas probabilidades para associar o vetor de probabilidades ao conjunto de hipóteses diagnósticas. A hipótese com maior probabilidade de ocorrência pode ser considerada a conclusão do sistema, note que esta conclusão está associada ao grau de certeza da resposta do sistema [NASSAR 98].

O comportamento de uma rede pode ser considerado como sendo um processo estocástico, e seus estados e sua evolução podem ser modelados utilizando a teoria da probabilidade. Portanto, torna-se relevante investigar a adequação do enfoque bayesiano para desenvolver um sistema especialista de apoio ao gerenciamento da rede.

2.2. Trabalhos Correlatos

Um sistema de gerência pró-ativa pode ser associado com a área de sistemas especialistas, podendo assim ser complementado com técnicas de inteligência artificial. Diversas técnicas de inteligência artificial estão sendo pesquisadas e aplicadas na área de

gerência pró-ativa. Como em [FRANCESCHI 97], que através dos dados colhidos por monitoração remota desenvolveu um sistema de gerência proativa com uso de inteligência artificial.

O sistema especialista mencionado em [ROCHA 97], utilizou os métodos de classificação de fichas e de construção de árvores de conhecimento. Estas árvores associam diagnósticos a seus fatores determinantes, que são ordenados conforme a importância e associados à sua relevância. Desta forma foi construído um conjunto de árvores que representou a base de conhecimento. Destas árvores foram derivadas as regras que compõe a base de regras do sistema. A implementação deste sistema foi feita em Prolog [CLOCKSIN 84], e teve a sua máquina de inferência implementada sobre o mecanismo de resolução do próprio Prolog.

Em [NETO 98] é proposto um modelo de gerenciamento pró-ativo que se utiliza de informações sobre as aplicações que estão trafegando em segmentos monitorados, foi aplicada a técnica de séries temporais [CHATFIELD 84] na análise do perfil de funcionamento destes segmentos e foi implementada uma série de funções de gerenciamento que utilizam de tal técnica e do nível de informações disponibilizadas pelo modelo proposto.

Técnicas de inteligência artificial foram também utilizadas no reconhecimento de problemas da rede por [ROCHA 97] e por [SÁENS 96].

Técnicas de *Data Mining* foram aplicadas sobre uma base de dados com informações relativas ao desempenho da rede de uma companhia aérea, para analisar e codificar os dados em conhecimento utilizável. Neste trabalho [KNOBBE 97] mostra que o conhecimento adquirido contribui para facilitar o trabalho de gerencia da rede.

A pesquisa de [KOEHLER 98] realizou um estudo sobre o raciocínio bayesiano em Sistemas Especialistas, tendo como domínio de aplicação a área médica. Foi desenvolvida uma aplicação para auxiliar na avaliação do estado nutricional em crianças com até 2 anos de idade com base nos sinais e sintomas e dados antropométricos. Foi testada a sensibilidade do sistema às mudanças nos valores de probabilidades e foi analisado o comportamento do sistema em função do tamanho da base de conhecimento.

Uma das conclusões desta pesquisa foi que na análise do tamano da base de conhecimento, o sistema se manteve estável até o ruído de 30% nas probabilidades condicionais e na análise do comportamento percebeu-se que as variáveis acrescentadas à base não trouxeram informações relevantes no processo de diagnóstico, isto é, neste caso o sistema era capaz de realizar diagnóstico considerando poucas entradas.

CAPÍTULO III

3. Fundamentação Teórica

Neste capítulo apresenta-se algumas considerações a respeito de probabilidade bayesiana, *data mining* e gerenciamaneo de redes de computadoes, julgadas relevantes para o desenvolvimento desta pesquisa.

3.1. Probabilidade Bayesiana

Os métodos Bayesianos possibilitam representar numericamente o grau de certeza sobre condições de incerteza, e manipulá-lo de acordo com as regras definidas na teoria da probabilidade, pois a teoria Bayesiana está fundamentada nesta teoria [HECKERMAN 95].

A Estatística Bayesiana tem origem no nome do matemático inglês Thomas Bayes. O teorema de Bayes é de grande importância para o cálculo de probabilidades. Mais especificamente, na área de interesse deste trabalho, o teorema de Bayes é o mecanismo fundamental que permite relacionar diversas probabilidades conhecidas pela monitoração da rede e deduzir uma probabilidade sintética, que com melhor eficácia estime o resultado.

O teorema de Bayes é um método quantitativo para a revisão de probabilidades conhecidas, com base em nova informação amostral. No processo de tomada de decisão, isto significa calcular uma probabilidade, pela aplicação de um teste diagnóstico (probabilidade *a posteriori*), considerando uma probabilidade já disponível (probabilidade *a priori*) [KOEHLER 98].

3.1.1. Espaço de Probabilidade

Seja ε um conjunto finito e **P** uma função de ε para os números reais não negativos, tal que:

$$\Sigma\, \mathbf{P}(e) = 1 \quad \forall\, \mathbf{e}: \mathbf{e} \in \varepsilon$$

O par **(ε,P)** é chamado de espaço de probabilidade. Os elementos de ε são chamados de eventos simples ou elementares. **P** é chamado de distribuição de probabilidade ou função de probabilidade. Desta definição resulta que a probabilidade de qualquer evento **e**, denotada por **P(e)**, é medida por um número no intervalo **[0;1]**.

Intuitivamente, ε é a coleção de resultados que se pode esperar em um domínio de aplicação. O valor **P(e)** é uma estimativa da crença de que o resultado **e** ocorra.

3.1.2. Probabilidade Condicional

O conceito de probabilidade condicional permite considerar as novas informações de forma a obter as novas probabilidades.

Sejam **A** e **B** eventos compostos de um espaço de probabilidades **(ε ,P)**. Suponha que um evento simples **e** ocorra. A probabilidade **P(B)** é a probabilidade de que $\mathbf{e} \in \mathbf{B}$ dado nosso conhecimento inicial refletido por **P**. Intuitivamente, **P (B|A)** é a probabilidade que $\mathbf{e} \in \mathbf{B}$ quando se tem a informação adicional de que $\mathbf{e} \in \mathbf{A}$. Seja **(ε, P)** um espaço de probabilidade e seja $\mathbf{A} \subseteq \varepsilon$ tal que $\mathbf{P(A)} \neq \mathbf{0}$. Define-se o espaço de probabilidade **(ε, f)** da seguinte forma:

$\mathbf{P(e) / P(A)}$ se $\mathbf{e} \in \mathbf{A}$ $\qquad$ $\mathbf{f(e) = 0}$ se $\mathbf{e} \notin \mathbf{A}$

Para qualquer $\mathbf{B} \subseteq \varepsilon$ a probabilidade condicional de **B** dado a ocorrência de **A** é igual a **f(B)**. Observe que neste caso **A** é o novo espaço de probabilidade, onde **B** deve ser analisado.

Se $\mathbf{A} = \varepsilon$ então $\mathbf{P(B|A) = P(B)}$.

3.1.3. Propriedades da Probabilidade Condicional

Seja um espaço de probabilidade $(\varepsilon, \mathbf{P})$. Se $\mathbf{C} \subseteq \varepsilon$ então $\mathbf{P(C) \neq 0}$. Segue-se que:

$\mathbf{P(A|C) = P(A \wedge C) / P(C)}$. Onde $\mathbf{P(A \wedge C)}$ significa a probabilidade de que ambos os eventos ocorram; isto é a probabilidade do evento **A** ocorrer **e** do evento **C** ocorrer.

Se $\mathbf{A \subseteq B \subseteq \varepsilon}$ então $\mathbf{0 \leq P(A|C) \leq P(B|C) \leq 1}$

Se $\mathbf{A, B \subseteq \varepsilon}$ então $\mathbf{P(A|C) = P(A \wedge B \mid C) + P(A \wedge \bar{B} \mid C)}$ e

$$\mathbf{P(A \vee B \mid C) = P(A|C) + P(B|C) - P(A \wedge B \mid C)}$$

Se $\mathbf{A_i \subseteq \varepsilon}$ para $\mathbf{1 \leq i \leq n}$ e $\mathbf{A_i \cap A_j = \varnothing}$ então para todo $\mathbf{i \neq j}$

$$\mathbf{P(A_1 \vee A_2 \vee ... \vee A_n|C) = P(A_1|C) + P(A_2|C) + ... + P(A_n|C)}$$

Onde $\mathbf{P(A_1 \vee A_2)}$ significa a probabilidade de que pelo menos um dos eventos ocorre; isto é a probabilidade do evento $\mathbf{A_1}$ ocorrer **ou** do evento $\mathbf{A_2}$ ocorrer.

Se $\mathbf{A \subseteq \varepsilon}$, $\mathbf{B_1 \vee B_2 \vee ... \vee B_n}$ para $\mathbf{1 \leq i \leq n}$ e $\mathbf{B_i \cap B_j = \varnothing}$ para todo $\mathbf{i \neq j}$ então

$$\mathbf{P(A) = P(A|B_1).P(B_1) + P(A|B_2).P(B_2) + ... + P(A|B_n).P(B_n)}$$

3.1.4. Teorema de Bayes

Seja o espaço de probabilidade (ε, P) e os eventos compostos $e, H_1, H_2, \ldots, H_K \subseteq \varepsilon$, desde que nenhum desses eventos tenha probabilidade nula, então:

$$P(H_i|e) = \frac{P(e|H_i) \cdot P(H_i)}{P(e)}$$

Se $P(H_i \wedge e) \neq 0$ para todo i então:

$$\frac{P(H_i|e)}{P(H_j|e)} = \frac{P(H_i)}{P(H_j)} \cdot \frac{P(e|H_i)}{P(e|H_j)}$$

Se os eventos $H_1 \cup H_2 \cup \ldots \cup H_k = \varepsilon$ e $H_i \cap H_j = \varnothing$ para todo $i \neq j$ então:

$$P(e) = P(H_1) \cdot P(e|H_1) + P(H_2) \cdot P(e|H_2) + \ldots + P(H_k) \cdot P(e|H_k)$$

Resultando:
$$P(H_i|e) = \frac{P(e|H_i) \cdot P(H_i)}{\sum_{j=1}^{k} (P(H_j) \cdot P(e|H_j))}$$

Nas aplicações dos sistemas especialistas probabilísticos os **Hi**'s são as hipóteses concorrentes. O evento **e** pode ser visto como uma evidência. O conhecimento da ocorrência desta evidência leva a mudanças na probabilidade *a priori* **P(Hi)** para a probabilidade condicional **P(Hi |e)**, que por sua vez considera a evidência **e**.

3.1.5. Independência de Eventos

Seja um espaço de probabilidade **(ε, P)**. E, sejam os eventos $e_1, e_2 \subseteq \varepsilon$. Segue-se que:

Se $P(e_1 \wedge e_2)=P(e_1).P(e_2)$ então os eventos e_1 e e_2 são independentes.

Genericamente, para qualquer subconjunto $E = \{ e_{i1}, e_{i2}, ..., e_{ik}\}$ de $\{ e_1, e_2, ..., e_n\}$ se $P(e_{i1} \wedge e_{i2} \wedge ... \wedge e_{ik} | H) = P(e_{i1} | H).\ P(e_{i2} | H) ... P(e_{ik} | H)$ então pode-se dizer que os eventos e_i's são eventos mutuamente independentes dado a hipótese **H**. A idéia básica do conceito probabilístico de independência é que o conhecimento de certa informação não traz informação adicional sobre outra coisa. Isto é, se e somente se, saber que o evento e_1 ocorreu não trouxer informação sobre o evento e_2, e saber que o evento e_2 ocorreu não trouxer informação sobre o evento e_1 então diz-se que ocorre a independência entre estes eventos.

3.1.6. Atualização Bayesiana para uma Nova Evidência

Seja **H** uma hipótese e $e^n = e_1, e_2, ..., e_n$ uma sequência de dados independentes observados no passado e seja **e** um novo fato. A probabilidade condicional para a nova evidência é dada por:

$$P(H|e^n \wedge e) = P(H \wedge e^n \wedge e) / P(e^n \wedge e) = (P(e^n) . P(H|e^n) . P(e|e^n \wedge H)) / ((P(e^n) . P(e|e^n))$$

resultando em: $P(H | e^n \wedge e) = P(H| e^n).((P(e| e^n \wedge H)/P(e| e^n))$

O resultado acima mostra que uma vez calculada a probabilidade condicional da hipótese **H** dado o conjunto e^n de evidências, isto é o valor $P(H|e^n)$, os dados passados e^n podem ser desprezados e assim pode ser obtido o impacto da nova evidência **e**. A crença velha $(H|e^n)$ assume o papel de crença *a priori* no cálculo do impacto da nova informação **e**; a probabilidade $P(H|e^n)$ sumariza completamente a experiência passada e para sua atualização necessita somente ser multiplicada pela *LIKELIHOOD ratio* $P(e| e^n \wedge H)$. Esta

razão mede a probabilidade do novo dado **e** considerando a hipótese **H** e os dados passados $\mathbf{e^n}$.

Geralmente, adota-se que a *LIKELIHOOD ratio* é independente dos dados passados e considera somente a nova evidência **e**.

A natureza incremental do processo de atualização para a nova evidência **e** pode ser explorado utilizando a razão *ODDS*:

$$\mathbf{O(H \mid e^n \wedge e) = O(H \mid e^n).L(e|H)} \qquad \mathbf{Log\ O(H \mid e^n \wedge e) = log\ O(H \mid e^n) + log\ L(e|H)}$$

Assim, o logaritmo da *LIKELIHOOD ratio* da evidência **e** pode ser visto como um peso da própria evidência **e**. Caso a evidência **e** suporte a hipótese **H** então terá um peso positivo, se for oposta a **H** então terá um peso negativo. Atualizar recursivamente as medidas de crenças está fortemente relacionada ao conceito de independência condicional [NASSAR 98].

3.2. *Data Mining*

Com a evolução da tecnologia e a complexidade dos dias atuais, a organização das informações para auxiliar a tomada de decisões é uma arma fundamental no arsenal das instituições e empresas competitivas.

Data mining (DM) é um dos campos da computação que está desenvolvendo-se rapidamente. Inicialmente ele era apenas uma pequena área de pouco interesse dentro da ciência da computação, mas tem se expandido rapidamente tornando-se uma área de pesquisa própria, em crescente desenvolvimento.

Embora possam ser vistas descrições de técnicas de *data mining* sendo expostas para descoberta de conhecimento, ou relatórios de aplicações de sucesso, provavelmente ainda resta uma dúvida básica: Por que *data mining* é útil ?

Procura-se aqui esclarecer algumas questões básicas sobre *data mining*, apresentando uma visão geral das técnicas utilizadas e alguns exemplos de aplicações.

As técnicas de *data mining*, quando aplicadas corretamente sobre uma base de dados preparada, podem revelar padrões e descobrir tendências relevantes para empresas em qualquer ramo de negócio, contribuindo de forma extraordinária para a tomada de decisões.

3.2.1. Razões para o crescimento da popularidade de DM

A Era do Conhecimento já iniciou-se. A importância de colecionar dados que refletem as atividades empresariais ou científicas para alcançar vantagens competitivas é hoje amplamente reconhecida. Sistemas poderosos para coletar e administrar grandes bancos de dados estão em todas as grandes e médias empresas. Porém, existe uma sub-utilização destes sistemas devido a dificuldade de extrair conhecimento a partir dos dados oferecidos.

A principal razão para a necessidade de automatizar sistemas para realizar análise de dados inteligentemente é o grande volume de novas informações que requerem processamento. A quantidade de dados acumulados a cada dia pelas várias empresas, comerciais, científicas e organizações governamentais, é muito grande. A grande quantidade de informações ocasiona a grande dificuldade da análise realizada apenas por humanos.

Outros problemas são apresentados pela análise humana, pois os processos de análise, quando procuram por complexas dependências nos dados, são inadequados para o cérebro humano. Um especialista humano tem sempre uma expectativa quando investiga um sistema. As vezes isto ajuda, as vezes atrapalha, mas isto pode dificultar o estabelecimento de regras e fatos imparciais.

Um benefício adicional do uso de técnicas automatizadas de *data mining* é que o processo tem um custo muito baixo se comparado com processos manuais, que exigem grande treinamento (e pagamento) de profissionais. Enquanto *data mining* não elimina a participação humana para resolver as tarefas completamente, ele simplifica significantemente o processo de análise e permite o trabalho de um analista que não é um profissional em estatística ou programador, gerenciando os processos para extração do conhecimento a partir dos dados [MEGAPUTER 00].

3.2.2. O que é *data mining*?

Para explicar o que *data mining* realmente é, este trabalho apresenta um conjunto de fatos introdutórios:

- Nas décadas mais recentes, muitas organizações tem gasto recursos consideráveis para a construção e manutenção de grandes bases de dados, incluindo o desenvolvimento em larga escala de sistemas de informação gerenciais.

- Em muitos casos a informação contida nestes bancos de dados não é totalmente utilizada porque os dados não podem ser facilmente acessadas ou analisados.

- Alguns bancos de dados tornaram-se tão grandes que até mesmo os administradores dos sistemas não sabem sempre qual informação pode ser representada ou quão relevante ela pode ser para resolver as questões que tem em mãos.

- Poderia ser um beneficio para organizações ter um caminho para "minerar" esta grande quantidade de informações, para selecionar informações importantes ou padrões que podem estar contidos nelas.

Por exemplo, são frequentes perguntas deste tipo:

- Que bens deveriam estar em promoção para determinado cliente?
- Qual é a probabilidade de um certo cliente responder a uma promoção?
- Negar ou fornecer um empréstimo a um cliente?
- Qual diagnóstico médico deve ser dado a um paciente?
- Por que, de repente, uma máquina começou a produzir peças defeituosas?
- Qual o tráfego esperado de uma determinada porta de um roteador?

Todas as perguntas acima poderiam ter sido respondidas se a informação escondida entre megabytes de informações, em seu banco de dados, pudesse ser encontrada explicitamente e pudesse ser utilizada. As ferramentas do *data mining* trabalham modelando o sistema investigado e descobrindo relações que ligam variáveis em um banco de dados[MEGAPUTER 00].

Sistemas modernos de *data mining* aprendem por si mesmos a partir de um prévio histórico do sistema investigado, formulando e testando hipóteses sobre as regras que este sistema obedece. Quando um conhecimento conciso, valioso e de interesse para o sistema é descoberto, ele pode e deve ser incorporado no processo de tomada de decisões e pode ajudar o administrador a tomar decisões mais inteligentes baseadas nas informações extraídas.

Existem muitas tecnologias e ferramentas disponíveis para aplicações em *data mining*. Certas tecnologias tem melhor desempenho que outras em termos de facilidade de uso e retorno do investimento, porém na maioria das vezes uma ferramenta sozinha nunca apresentará uma solução completa.

A idéia geral é que fora as situações em que se possa empregar métodos matemáticos padrão ou análise estatística para testar hipóteses pré-definidas, *data mining* é

a solução mais útil na análise exploratória em cenários onde não existem noções predeterminadas sobre o que irá constituir um saída interessante. *Data mining* é um processo iterativo onde o progresso é definido pela descoberta, quer seja automática ou manual [WESTPHAL 98].

Existem muitas definições diferentes sobre *data mining*: segundo Piatetsky-Shapiro, [MEGAPUTER 00], "*Data Mining* é o processo de identificar conhecimento válido, recente, potencialmente útil e compreensível a partir de banco de dados, que é usado para tomar decisões cruciais nas empresas".

Segundo [FAYYAD 96], *data mining* é uma etapa do processo de *Knowledge Data Discovery* consistindo de algoritmos particulares de mineração de dados que, sob algumas limitações de eficiência computacionais aceitáveis, produz uma enumeração particular de padrões sobre uma base de dados.

3.2.3. O que é *Knowledge Data Discovery*?

Knowledge Data Discovery (KDD) não é uma nova técnica mas sim um conjunto de tecnologias que envolvem *Machine Learning*, estatística, tecnologias de bancos de dados, sistemas especialistas e visualização de dados.

Machine learning é o estudo de métodos computacionais, utilizados para melhorar o desempenho de sistemas de aprendizagem, através da automação da aquisição de conhecimento a partir de experiências. *Machine learning* busca aumentar o nível de automação no processo de engenharia de conhecimento, substituindo o tempo gasto pelas atividades executadas por humanos por técnicas automáticas, que provém eficiência e eficácia na descoberta e exploração de conhecimento; podendo criar automaticamente a base de conhecimento requerida pelos sistemas especialistas [LANGLEY 95].

O processo de KDD não requer que os usuários criem hipóteses sobre os relacionamentos e correlações entre variáveis, é possível ao usuário, por exemplo, simplesmente perguntar quais são as variáveis que afetam as vendas de um produto específico. Em todo o processo de KDD o *Data Mining* representa 20 % os outros 80 % são atribuídos à preparação dos dados [MEGAPUTER 00].

Várias definições são aplicadas para KDD, como por exemplo: é o processo não trivial de identificar padrões nos dados válidos, novos, potencialmente úteis, e compreensíveis [FRAWLEY 91].

Para [FAYYAD 96], KDD é o processo de, usando métodos (algoritmos) de mineração de dados, extrair (identificar) conhecimento, de acordo com o que é julgado conhecimento pelas métricas e saídas esperadas, usando uma base de dados onde é requerido algum preprocessamento, subexemplificação e transformações.

Para o processo de KDD ser realizado com sucesso, são necessários seguir alguns passos fundamentais [MEGAPUTER 00]:

- **Definição de Objetivos** – Primeiramente deve-se definir o objetivo em termos de negócio, ou seja, qual o retorno esperado, o que deseja-se atingir, como por exemplo, definir o que utilizar em uma promoção com mala direta, qual o segmento do mercado consumidor a atingir. Depois define-se os objetivos do próprio KDD, como por exemplo identificar clientes de alto risco, identificar afinidades entre os clientes, como por exemplo, quais os clientes que jogam tênis. E, finalmente, define-se o que fazer com os resultados: alterar o plano de mercado, criar campanha promocional.

- **Seleção de Dados** – Deve-se definir o volume adequado de dados, se existirem muitos dados o processo será mais demorado, por outro lado, poucos dados implicariam na falha de conclusão. A base de dados deve ser trabalhada de forma a eliminar as colunas[2] inúteis, como por exemplo, o nome dos clientes, quando você

[2] Excluir uma determinada coluna corresponde a exluir uma determinada variável de todos os registros.

tenta definir o perfil de seu consumidor. Deve-se também exlcuir as linhas[3] com valores muito errados, isto poderia alterar a conclusão do processo de KDD.

- **Entendimento dos Dados** – Pode-se utilizar diferentes ferramentas para compreender uma base de dados: Ferramentas de Query que podem determinar valores estatísticos de colunas relevantes; Ferramentas Estatísticas que podem por exemplo traçar a média e sua variação nos dados de uma coluna. Ferramentas OLAP que podem determinar relações entre colunas e linhas; Ferramentas de Visualização que mostram em gráficos e imagens bidimensionais ou tridimensionais as informações da base de dados.

- **Limpeza dos Dados** – Deve-se aqui verificar, por exemplo, a duplicação de casos ou registros na base de dados, a plausibilidade das informações contidas na base de dados.

- **Enriquecimento dos Dados** - Descobertas as falhas, pela limpeza dos dados, deve-se enriquecer a base de dados, trocando os dados falsos por verdadeiros. Por exemplo, no caso de existir três códigos diferentes para o mesmo cliente devido aos nomes digitados de forma diferente: Cleverson A. Veronez; Cleverson Veronez e Cleverson Alessandro Veronez. Deve-se verificar qual o nome correto e alterar os registros.

- **Preparação dos Dados** – Dependendo da técnica de DM utilizada, os dados devem ser preparados de formas diferentes; alguns passos podem ser necessários como por exemplo realizar cálculos, agrupar valores, transformar o conteúdo de determinado campo, nivelamento de valores, etc.

- **Criação do Modelo para DM** – Deve ser criado o modelo levando em conta duas diferentes abordagens: o processo de verificação, que será realizado baseado nas hipóteses do usuário; e, o processo de descoberta, realizado automaticamente através de machine learning.

---

[3] Excluir uma determinada linha corresponde a excluir um determinado registro, um determinado caso coletado.

- **Data Mining** – Deve-se definir as técnicas de *data mining* a serem utilizadas no processo de descoberta. Uma descrição suscita de algumas técnicas de *data mining* será vista posteriormente.

- **Monitoração do Modelo** - O modelo criado deve ser constantemente monitorado pois as características dos dados podem mudar e surgem diariamente novos dados para serem processados. Outra razão importante para se monitorar o modelo é que novas oportunidades podem surgir com uma visão mais detalhada dos dados. O modelo pode ser monitorado e validado por um especialista humano.

3.2.4. Utilização

Um dos usos mais comuns do *data mining* comercialmente são em aplicações do tipo *Market basket*. Estas aplicações procuram por padrões entre as vendas de dois produtos relacionados diretamente ou não, buscando relações entre os perfis dos consumidores.

Utilizando aplicações do tipo *market basket*, a Wal-Mart descobriu que o perfil do consumidor de cervejas é semelhante ao de fraldas. Eram homens casados entre 25 e 30 anos, que compravam fraudas e/ou cervejas à tarde no caminho do trabalho para a casa. Então as fraudas e cervejas foram colocadas lado a lado nos pontos de venda e o consumo cresceu 30% às sextas-feiras [MEGAPUTER 00].

Um outro exemplo, um software chamado Advanced Scout desenvolvido pela IBM, fornece aos treinadores de basquete a possibilidade de descobrir muitos fatos desconhecidos apesar de todas as estatísticas providas pela NBA. Em um jogo entre NY Knicks e Charlote Horners foi constatado que Glenn Rice converteu 83% dos jumps arremessados de uma determinada posição. Esta informação foi considerada relevante pelo software pois a média de acertos do Charlote durante o jogo foi de 54%. Isto mostra que

esta é uma jogada que deve ser mais explorada pelo Charlote e do ponto de vista dos seus adversários que Glenn Rice deve ser muito bem marcado nesta posição [MEGAPUTER 00].

As técnicas de *data mining* podem ser utilizadas em muitas situações [MEGAPUTER 00]:

1. **Varejo / Marketing**
   - Market Basket
   - Características demográficas entre clientes
   - Resposta para mala direta
   - Padrões de comportamento entre clientes

2. **Bancos**
   - Fraudes na utilização de cartões de crédito
   - Identifica clientes leais
   - Prognóstico dos clientes com tendências de mudar de cartão de crédito
   - Determina perfil de gastos por grupo
   - Descobre correlações entre indicadores econômicos

3. **Telecomunicações**
   - Determinar novas políticas de serviços
   - Análise detalhada de chamadas

- Análise de mercado

4. **Seguros**

- Análise de risco
- Fraudes
- Perfil de segurados com sinistro

5. **Saúde**

- Identificar tratamentos de doenças
- Características de pacientes
- Determinação de grupo de risco

6. **Governo**

- Fraude
- Planejamento de Orçamento
- Análises econômicas
- Gerência de Recursos Humanos

3.2.5. Tecnologias e sistemas

Em termos de tecnologia de implementação, as ferramentas podem ser classificadas em dois grupos, um que trabalha com análise de dados retidos, ou seja, os

dados analisados são armazenados para futuras comparações; e busca de padrões, onde os dados são deixados de lado depois de utilizados pela ferramenta [MEGAPUTER 00].

Seria muito instrutivo antes de discutir as várias ferramentas existentes para data mining considerar três critérios vitais [FAYYAD 96]:

- Controle de significância dos resultados obtidos;
- Transparência do desenvolvimento de modelos empíricos e sua interpretabilidade;
- Grau do processo de pesquisa automatizado e sua facilidade de utilização.

Para construir uma ponte entre os métodos mais tradicionais de análise de dados e os métodos de *data mining* este trabalho começa pela discussão de alguns métodos mais tradicionais como sistemas analíticos orientados ao assunto e pacotes estatísticos, e então considera outros métodos: redes neurais, programação evolucionária, raciocínio baseado em casos, árvores de decisão, algorítmos genéticos e métodos de regressão não-linear.

3.2.5.1. Sistemas analíticos orientados ao assunto

Todos os sistemas são, com certeza, domínio de aplicação específico. Como é um dos sistemas mais desenvolvidos deste tipo, aqui são considerados sistemas para análise de mercados financeiros baseados no método de análise técnica. Análise técnica representa um pacote de algumas dúzias de técnicas diferentes para prever a dinâmica de preços e selecionar a estrutura ótima para a pasta de investimento, baseado em vários modelos empíricos do comportamento de mercado [MEGAPUTER 00].

O modelo empírico subjacente é implantado manualmente em cada sistema, em lugar de ter sido derivado por aprendizagem automática. Assim, os dois primeiros critérios a considerar são: consideração do significado estatístico dos modelos derivados e sua interpretabilidade.

Tais sistemas normalmente provêem outra vantagem para o usuário. Eles operam em condições específicas do campo de aplicação. Estas condições estão claras aos comerciantes e aos analistas financeiros. Freqüentemente tais sistemas têm interfaces especiais devido ao fato de carregar dados financeiros. Existe um grande número de sistemas analíticos orientados ao assunto baseados em análise técnica.

Exemplos de sistemas e/ou *Shells*:

- MetaStock (Equis International, USA)
- SuperCharts (Omega Research, USA)
- Candlestick Forecaster (IPTC, USA)
- Wall Street Money (Market Arts, USA)

3.2.5.2. Pacotes Estatísticos

Enquanto que na maioria das versões mais recentes dos pacotes estatísticos, métodos estatísticos tradicionais bem conhecidos são completados por alguns elementos de *data mining*, os principais métodos de análise de dados principais continuam sendo de natureza clássica: correlação, regressão, análises fatorial e outras técnicas. Tais sistemas não podem determinar as dependências escondidas nos dados e requerem que o usuário explicite suas próprias hipóteses, que serão testadas pelo sistema.

Uma das principais desvantagens de tais sistemas é que eles não permitem que um usuário sem um treinamento completo em estatística execute a análise de dados. Um usuário tem que passar por alguns meses de cursos especiais para poder usar mais ou menos inteligentemente estes sistemas. Outra desvantagem deste método é que durante a exploração de dados o usuário tem que executar um conjunto de algumas operações elementares repetidamente. Ferramentas para automatização do processo muitas vezes não existem, ou requerem programação em algum idioma interno.

Neste trabalho são utilizadas algumas técnicas estatísticas, dentre elas a aplicação da probabilidade bayesiana, que tem grande destaque no processo de DM.

Exemplos de *softwares* estatísticos e/ou *Shells*:

- *Statistical Analysis System* (SAS Institute, USA)
- *Statistical Package for Social Science* (SPSS, USA)
- Statgraphics (Statistical Graphics, USA)
- Statistica (Statsoft, USA)

3.2.5.3. Redes Neurais

Informalmente uma rede neural artificial é um sistema composto por vários neurônios de modo que as propriedades de sistemas complexos sejam usadas. Estes neurônios estão ligados por conexões, chamadas conexões sinápticas [BARRETO 97].

Esta é uma classe grande de diversos sistemas cuja arquitetura imita estrutura do tecido neural vivo, até certo ponto construída por neurônios separados. Uma das arquiteturas mais difundidas, *multilayered perceptron with back propagation*, emula o trabalho de neurônios incorporados em uma rede hierárquica, onde cada neurônio de um nível é conectado com as saídas de todos os neurônios do nível anterior. Os dados são tratados e analisados como parâmetros de excitação dos neurônios, vistos como verdadeiro alimento para os neurônios do primeiro nível. Estas excitações são propagadas a partir dos neurônios do primeiro nível para os do próximo nível, sendo ampliado ou debilitado de acordo com pesos (coeficientes numéricos) designados para as conexões intraneurais correspondentes. Como resultado final deste processo, um único neurônio detém o conhecimento dos neurônios de nível mais alto e adquire um pouco de valor (força de excitação), sendo considerado uma predição – a reação da rede inteira para os dados processados [MEGAPUTER 00].

Para fazer predições significantes, primeiramente uma rede neural tem que ser treinada com dados que descrevem situações prévias, devem ser introduzidos parâmetros e reações corretas para os neurônios. Treinamento consiste em selecionar pesos designados às conexões, o que provêem a proximidade maximal das reações produzidas pela cadeia de neurônios às reações corretas conhecidas [MEGAPUTER 00].

Esta aproximação provou ser efetiva em problemas de reconhecimento de imagem. Porém podem não ser tão efetivas em aplicações que exigem maior grau de certeza, como aplicações médicas ou financeiras. Existem várias razões para esta dificuldade. Primeiramente, as redes neurais construídas para analisar um sistema complexo, como mercados financeiros, também são redes complexas. Elas incluem dúzias de neurônios com centenas de conexões entre eles. Como resultado, o número de graus de liberdade do modelo (estes são os pesos de todas as conexões entre os neurônios da cadeia) freqüentemente fica maior que o número de exemplos (dados registrados separadamente) usados para treinar a rede. Isto priva o modelo de qualquer possibilidade de previsão correta [MEGAPUTER 00].

A segunda desvantagem é a não-transparência das redes neurais. Esta dificuldade também é vista de perto devido a complexidade da estrutura da rede neural: o conhecimento refletido em termos de pesos das conexões não podem ser analisadas e interpretadas por um ser humano.

Deveria ser notado que apesar das dificuldades, as redes neurais são ativamente usadas (com sucesso variado) em diferentes aplicações, principalmente em aplicações financeiras na maioria dos países desenvolvidos.

Exemplos de sistemas e/ou *Shells*:

- PolyAnalyst (Megaputer Intelligence, Rússia)
- BrainMaker (CSS, USA)
- NeuroShell (Ward Systems Group, USA)
- OWL (Hyperlogic, USA)

3.2.5.4. Programação Evolucionária

No momento este é o método mais jovem e evidentemente a filial mais promissora do *Data Mining*. A idéia subjacente do método é que o sistema formule hipóteses automaticamente, sobre a dependência da variável em relação à outras variáveis, na forma de programas expressos em uma linguagem de programação interna. Utilizando uma linguagem de programação universal a aproximação assegura que qualquer dependência ou algoritmo podem ser expressos, em princípio, nesta linguagem [MEGAPUTER 00].

O paradigma evolucionário de desenvolvimento de programas, é aquele em que não se consegue especificar exatamente o que se deseja antes de começar a programar. Segue a inspiração da Natureza segundo a teoria da evolução, em que não sabe-se onde vai chegar [BARRETO 97].

O processo de produção de programas internos (hipóteses) é organizado como evolução no mundo de todos os possíveis programas. Quando um programa encontra uma hipótese que descreve razoavelmente bem a dependência observada, começa a apresentar várias modificações leves a este programa e seleciona os melhores programas filhos, alcançados por este processo, assim melhora-se a precisão da predição. Deste modo o sistema desenvolve várias linhas genéticas de programas, que competem entre si procurando expressar com maior precisão a dependência procurada [MEGAPUTER 00].

Quando o melhor programa (hipótese) com um nível de precisão desejado é obtido, um módulo especial do sistema traduz a dependência descoberta da linguagem interna para uma forma explícita entendida pelo ser humano: fórmulas matemáticas, tabelas, etc. Isto proporciona para o usuário uma perspicácia e controle da dependência obtida, bem como permite a visualização dos resultados [MEGAPUTER 00].

Exemplos de sistemas e/ou *Shells*:

- PolyAnalyst (Megaputer Intelligence, Russia)

3.2.5.5. Raciocínio Baseado em Casos (CBR)

A idéia principal deste método é muito simples. Para prever uma situação futura, ou para tomar uma decisão correta, os sistemas que utilizam este método encontram um caso análogo e escolhem a mesma solução que foi aplicada no caso anterior. Devido a esta razão este método é também chamado *nearest neighbor method.* Sistemas de CBR demonstram resultados bastante bons em problemas diversos. Por outro lado, uma grande desvantagem é que estes sistemas não criam qualquer modelo ou regras que resumem a experiência prévia. Suas predições estão baseadas em processar o conjunto inteiro de dados históricos disponíveis, e assim, para o CBR, é difícil contar quais fatores específicos influenciaram na predição do sistema [MEGAPUTER 00].

Exemplos de sistemas e/ou *Shells*:

- KATE tools (Acknosoft, France)
- Pattern Recognition Workbench (Única, USA)

3.2.5.6. Árvores de Decisão

Este método só pode ser aplicado para solução de tarefas de classificação. Isto limita aplicação da árvore de decisão em muitos campos. Por exemplo, em aplicações financeiras, onde o problema mais comum é a tarefa de predizer valores de alguma variável de cálculo numérico.

Como resultado da aplicação deste método temos uma estrutura hierárquica de regras de classificação do tipo " IF ...THEN...". Esta estrutura tem a forma de uma árvore, como a da figura 3.1. Para decidir para qual classe um objeto ou uma situação deveria ser mandado devem ser respondidas perguntas localizadas nos nós da árvore, a partir da raiz. Estas perguntas são da forma "o valor de variável Y é maior que x ?". Se a resposta é sim,

a pessoa segue a filial direita da árvore para um nó do próximo nível, se a resposta é não – a filial esquerda. E então a pergunta do próximo nó deve ser respondida, e assim por diante. Seguindo este procedimento chega-se a conclusão de qual classe o objeto considerado deve pertencer.

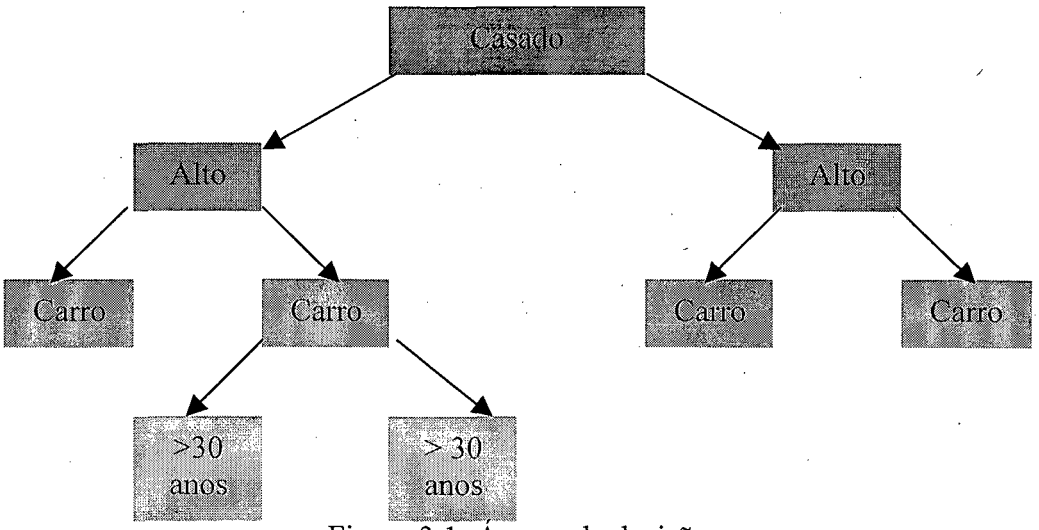

Figura 3.1: Árvore de decisão

Uma vantagem deste método é que esta forma de representação de regras é intuitiva e facilmente entendida pelo ser humano. Porém, a determinação do significado de uma regra torna-se um problema muito sério. O problema origina-se do fato que há um número menor de registros a cada nível da árvore de classificação que está sendo construída. A árvore está dividindo dados em um grande número de pequenos conjuntos de casos específicos. Se a árvore construída está bastante "fechada": se contém um número grande de filiais pequenas, então tal uma árvore não provê uma solução significante e estatisticamente justificada. Para aplicações em problemas complexos este método não acha uma solução satisfatória [MEGAPUTER 00] .

Existem muitos sistemas que implementam o método de árvore de decisão, tais como:

- C5.0 (Rule Quest, Austrália)

- Clementine (Integral Solutions, Great Britain)
- SIPINA (University of Lyon, France)
- IDIS (Information Discovery, USA)

3.2.5.7. Métodos de Regressão Não-Linear

Estes métodos estão baseados em procurar uma dependência da variável designada em outras variáveis na forma de funções predeterminadas. Por exemplo, em uma das implementações mais prósperas de algoritmos deste tipo: *group attribute accounting method*, uma dependência é buscada na forma de polinômios. Em princípio uma fórmula obtida, um polinômio, é mais satisfatória para análise e interpretação (na realidade é normalmente muito complexo). Assim este método tem chances melhores de prover soluções fidedignas em aplicações envolvidas com mercados financeiros ou diagnósticos médicos [MEGAPUTER 00].

Exemplos de sistemas e/ou *Shells*:

- PolyAnalyst (Megaputer Intelligence, Russia)
- NeuroShell (Ward Systems Group, USA)

3.2.6. Tarefas que são resolvidas utilizando DM

Existem várias técnicas e métodos que são usados como base para ferramentas de data mining. Estas técnicas e métodos podem desempenhar as seguintes tarefas [FAYYAD 96]:

Predição – aprender um padrão a partir de exemplos e usar o modelo desenvolvido para predizer futuros valores das variáveis selecionadas. É possível entrar com um registro de dados semi preenchido e esperar que a ferramenta de *data mining* preencha os valores que estão faltando, baseado nos padrões encontrados.

Classificação –encontrar uma função que mapeia um caso em uma dentre diversas classes discretas de classificação.

Detecção de relações –estabelecer relações entre variáveis.

Explicitação de modelos –encontrar fórmulas explícitas descrevendo dependências entre várias variáveis.

Clustering –identificar um conjunto finito de categorias ou setores para descrever dados.

Detecção de chaves –determinar as escolha mais significantes dentre algumas chaves obtidas nos dados a partir de previsões ou valores esperados.

Descoberta de padrões – é o processo onde padrões são procurados em uma base de dados sem ter idéia do que se espera encontrar. O conhecimento extraído da base de dados é geralmente colocado na forma de regras *IF THEN* ou na forma de associações, por exemplo: quando o evento 1 ocorre o evento 2 ocorrerá também.

Forensic analysis – é o processo de extrair elementos de dados que fogem dos padrões encontrados pelo processo de descoberta. Estes elementos podem significar uma exceção ao padrão.

3.3. Gerenciamento de Redes de Computadores

Através dos avanços das tecnologias de interconectividade e dos benefícios proporcionados pelas redes de computadores, cada vez mais computadores são interconectados nas organizações. Paralelamente, a diminuição dos custos dos

equipamentos permite adquirir e agregar à rede cada vez mais equipamentos, de tipos diversos, tornando essas redes cada vez maiores e mais complexas. Com isso, as redes começaram a ser interconectadas muito rapidamente; redes locais conectadas a redes regionais, as quais, por sua vez, ligadas a backbones nacionais.

Hoje essas redes são extremamente importantes para o dia-a-dia de muitas empresas em todo o mundo, porque normalmente, junto com sua utilização, vem a eficácia e a competitividade. Essa importância vem crescendo de tal forma que as empresas têm se tornado altamente dependentes destas redes, sentindo imediatamente o impacto quando os seus recursos não estão disponíveis.

Este novo ambiente originou alguns problemas administrativos. As tarefas de configuração, identificação de falhas e controle de dispositivos e recursos da rede passaram a consumir tempo e recursos das organizações.

Cientes desse problema, a solução então passou a ser buscada na atividade de Gerenciamento de Rede. Esta atividade passou a evoluir de forma rápida e concisa, sendo hoje uma das especialidades da área de redes que mais cresce. Apesar desses avanços, ainda hoje, é difícil conceituar esta atividade. Ao longo do seu desenvolvimento muitas definições foram propostas para a Gerência de Redes, resume-se a seguir algumas dessas definições [SAMPAIO 97]:

- Consiste no controle e administração de forma racional dos recursos de hardware e software em um ambiente distribuído, buscando melhor desempenho e eficiência do sistema.

- Consiste no controle de uma rede e seus serviços.

- Tem por objetivo maximizar o controle organizacional das redes, de maneira mais eficiente e confiável, ou seja, planejar, supervisionar, monitorar e controlar qualquer atividade da rede.

- Consiste na detecção e correção de falhas, em um tempo mínimo, e no estabelecimento de procedimentos para a previsão de problemas futuros.

3.3.1. Protocolos de Gerenciamento

É clara a necessidade de estabelecer monitoramento e controle sobre todos os componentes da rede, de forma a garantir que esta esteja sempre em funcionamento e que os problemas sejam identificados, isolados e solucionados o mais rápido possível. Entretanto, esta não é uma tarefa fácil. As redes têm assumido grandes proporções, com um grande número de computadores, além da constante adição de novos componentes, oferecendo integração dados/voz, multiplexadores e roteadores, além de tantos outros, o que tem adicionado mais complexidade a esse ambiente.

Para atender a esta necessidade de gerenciamento foram desenvolvidos protocolos de gerenciamento. A principal preocupação de um protocolo de gerenciamento é permitir aos gerentes de rede realizar tarefas, tais como: obter dados sobre desempenho e tráfego da rede em tempo real, diagnosticar problemas de comunicação e reconfigurar a rede atendendo às mudanças nas necessidades dos usuários e do ambiente. Porém, vários obstáculos teriam que ser superados, entre eles a heterogeneidade dos equipamentos de rede (computadores, roteadores e dispositivos de meio), dos protocolos de comunicação e das tecnologias de rede. Adicionalmente, é necessário que esse gerenciamento seja integrado, pois uma solução genérica e integrada auxilia os usuários a evitar os altos custos de uma solução específica, além de facilitar a manutenção, o monitoramento, o crescimento e a evolução da rede. Sem um gerenciamento integrado, a rede pode degradar até se tornar completamente ineficiente [SAMPAIO 97].

As informações de gerenciamento permitem, dentre outras tarefas, produzir registros de auditoria para todas as conexões, desconexões, falhas na rede e outros eventos significativos na rede. Esses registros, por sua vez, permitem determinar futuras necessidades de adicionar equipamentos, identificar e isolar erros comuns de um cliente, além de estudar outras tendências de uso. Deve-se ainda destacar que estas informações devem possibilitar uma atuação preventiva, e não meramente reativa, com relação aos problemas[SAMPAIO 97].

Ciente destas dificuldades, a ISO (*International Organization for Standardization*) vem desenvolvendo padrões para o gerenciamento de redes OSI (*Open Systems Interconnection*). Assim, de um lado temos o protocolo de gerência CMIP (*Common Management Information Protocol*), que segue o modelo da ISO. Do outro lado, existe o IETF (*Internet Engineer Task Force*) com um conjunto de padrões para o gerenciamento de redes TCP/IP (*Transmission Control Protocol / Internet Protocol*), normalmente referenciados como SNMP (*Simple Network Management Protocol*). Quase todas as plataformas de gerenciamento de redes Internet comercialmente disponíveis implementam o protocolo SNMP devido à sua simplicidade de implementação em relação ao CMIP [CARVALHO 93].

3.3.2. Áreas Funcionais da Gerência de Redes

O gerenciamento é uma prática vital para a operação das redes. O uso dos serviços das redes é afetado pela disponibilidade e eficiência do gerenciamento de redes. Atualmente, as atividades de controle e monitoração, quando disponíveis, em sua maioria, são realizadas por meio do uso de ferramentas ou softwares proprietários, cujo manuseio não é integrado. Isto torna o gerenciamento da rede caro e ineficiente. Assim sendo, a necessidade de produtos de gerenciamento que, independente do fabricante, interajam entre si é fundamental.

Com o objetivo de suprir estas necessidades, a ISO (*International Organization for Standardization*) desenvolve padrões de gerenciamento que permitem a interoperabilidade de múltiplos e diversificados sistemas computacionais e redes de computadores.

Dentro deste contexto, a ISO dividiu a atividade de gerenciamento em cinco áreas funcionais específicas (*SMFA's – Specific Management Functional Areas*): Gerenciamento de Configuração, Gerenciamento de Falhas, Gerenciamento de Contabilização,

Gerenciamento de Desempenho e Gerenciamento de Segurança. Dentro de cada Área Funcional, foram desenvolvidos padrões de funções (incluindo requisitos, modelos e serviços) para o gerenciamento das redes. Essas funções são processos de aplicação de gerenciamento que utilizam os serviços oferecidos pela camada de aplicação [SAMPAIO 97].

O SNMP, devido a sua natureza como uma solução rápida e reduzida do CMIP da ISO, implementa apenas parte das funções definidas pela ISO. Por isso ele é considerado "simples" de implementação e é um protocolo bem mais "leve" que o CMIP [SAMPAIO 97].

3.3.2.1. Gerenciamento de Configuração

A função do Gerenciamento de Configuração envolve a manutenção e monitoração da estrutura física e lógica da rede, incluindo a existência de componentes e sua interconectividade. Corresponde ao conjunto de processos que exercem o controle sobre os objetos gerenciados, identificando-os, coletando e fornecendo dados sobre os mesmos a fim de dar suporte a funções de [SAMPAIO 97]:

- atribuição de valores iniciais aos parâmetros do sistema;
- início e encerramento de operações sobre objetos gerenciados;
- alteração da configuração do sistema;
- associação de nomes a conjuntos de objetos gerenciados.

O objetivo principal do Gerenciamento de Configuração é manter um registro detalhado das configurações de rede antigas, atuais e propostas. Dependendo do ambiente, esse conhecimento detalhado pode compreender uma lista muito grande de informações sobre configuração.

Mudanças, inclusões e exclusões na rede devem ser acompanhadas pelo sistemas de gerenciamento para que sempre se conheça a configuração da mesma; isto é conseguido através do Gerenciamento de Configuração. Esta é provavelmente a parte mais importante

do gerenciamento de rede, pois não se pode gerenciar uma rede sem que se conheça ou se tenha acesso à configuração da mesma.

Documentar a configuração de rede irá auxiliar o administrador a entender os efeitos das alterações e falhas da mesma. Como todas as redes necessitam de manutenção, é fundamental ter uma visão completa das mesmas.

Através do Gerenciamento de Configuração é possível, por exemplo, controlar informações gerais sobre licença de software, computadores e outros dispositivos, bem como informações detalhadas sobre versão de aplicativo e *driver*.

3.3.2.2. Gerenciamento de Falhas

O Gerenciamento de Falhas é responsável pela manutenção e monitoração do estado de cada um dos objetos gerenciados e pelas ações necessárias ao restabelecimento ou isolamento das unidades com problemas. As informações coletadas podem ser usadas em conjunto com o mapa da rede, para indicar quais elementos da rede estão funcionando, quais operam precariamente ou quais permanecem fora de operação [SAMPAIO 97].

Também é possível que o Gerenciamento de Falhas gere um registro das ocorrências, um diagnóstico de falhas e uma associação entre os resultados do diagnóstico e as subsequentes ações de reparo.

O Gerenciamento de Falhas utiliza hardware e software para alertar os gerentes a respeito de uma falha e para auxiliar no reparo. Pode ser também utilizado hardware e software de tolerância a falhas ou redundantes que podem continuar a fornecer serviços de rede mesmo quando ocorrer falha.

Por exemplo, seria possível utilizar as seguintes ferramentas para realizar o gerenciamento de falhas [SAMPAIO 97]:

- ***Sistema de gerenciamento de rede*** – Um sistema de gerenciamento de rede é uma combinação de hardware e software que acompanha o funcionamento dos

componentes da rede. Este sistema geralmente inclui um terminal que concentra e emite os alarmes, uma indicação visual dos dispositivos que falharam e uma interface com o dispositivo de relatório remoto.

- ***Analisador de protocolo*** – Um analisador de protocolo é uma ferramenta de hardware e de software que monitora o tráfego na rede. Essa ferramenta pode ajudar a compreender as complexas interações que ocorrem na rede, identificando como os protocolos são utilizados em cada comunicação.

- ***Verificador de cabo*** – Um verificador de cabos é um dispositivo de hardware que identifica falhas no meio de transmissão. Dependendo do meio, pode ser identificado o cabo específico que está falhando e onde ocorreu a falha.

- ***Sistemas redundantes*** – Os sistemas redundantes utilizam peças idênticas de hardware ou de software para realizar as mesmas funções. Por exemplo, servidores de arquivos espelhados que armazenam exatamente os mesmos dados. Se um servidor falhar, o outro imediatamente o substitui continuando a servir os clientes da rede. Com isso, pode-se identificar o servidor com defeito e repará-lo sem que os usuários da rede sintam o impacto disto.

- ***Dispositivo de backup e arquivamento de dados*** – Os dispositivos de backup e o arquivamento de dados não ajudam a identificar falhas, mas podem reduzir significativamente o impacto das perdas causadas por falhas nos arquivos.

O número crescente de produtos de detecção e reparo de falhas que têm sido comercializados proporcionam sistemas de gerenciamento de falhas cada vez mais eficazes.

3.3.2.3. Gerenciamento de Contabilização

O Gerenciamento de Contabilização preocupa-se com a manutenção e monitoração de quais recursos e do quanto estes recursos estão sendo utilizados. Estas informações podem ser utilizadas para estatísticas ou para "faturamento".

As informações colhidas pelo Gerenciamento de Contabilização podem ser utilizadas para alocar novos recursos na rede ou simplesmente para planejar melhorias.

Utilizando o Gerenciamento de Contabilização, pode-se compreender os custos reais da rede, definir suas capacidades e estabelecer políticas e procedimentos para torná-la mais eficiente.

3.3.2.4. Gerenciamento de Desempenho

Enquanto o Gerenciamento de Falhas é principalmente reativo, o Gerenciamento de Desempenho é ativo, envolvendo a coleta e interpretação das medições periódicas dos indicadores de desempenho, identificando gargalos, avaliando tendências, e fazendo previsões do desempenho futuro da rede [SAMPAIO 97].

O Gerenciamento de Desempenho preocupa-se com o desempenho corrente da rede, incluindo parâmetros estatísticos tais como atrasos, vazão, disponibilidade e número de retransmissões. Consiste em um conjunto de funções responsáveis por manter e examinar registros com histórico dos estados do sistema para fins de planejamento e análise.

3.3.2.5. Gerenciamento de Segurança

Gerenciamento de Segurança refere-se à proteção e controle de acesso das informações de gerenciamento. Isso pode envolver a geração, distribuição e armazenamento de chaves de criptografia. Senhas, autorizações e outros controles de acesso devem ser mantidos de forma a proteger qualquer informação de gerenciamento de cada elemento da rede.

O Gerenciamento de Segurança aborda os aspectos de segurança essenciais para operar uma rede corretamente e proteger os objetos gerenciados. O sistema de

gerenciamento deve providenciar alarmes para o administrador da rede quando eventos relativos a segurança forem detectados.

A tarefa de Gerenciamento de Segurança pode abranger o seguinte [SAMPAIO 97]:

- Identificar os riscos de segurança e suas conseqüências.
- Implementar projetos e equipamentos de rede seguros.
- Administrar grupos e senhas de usuários.
- Usar equipamentos de monitoração da rede para registrar o uso, relatar violações ou fornecer alarmes para atividades de alto risco.

3.3.3. MIB (Management Informarion Base)

O conhecimento das MIB's (Base de Informações Gerenciáveis), e principalmente, o conhecimento de como utilizar estas informações, são de fundamental importância na Gerência de Redes.

Antes de definir o que é uma MIB, será introduzido o conceito de objetos gerenciados.

Um objeto gerenciado é a visão abstrata de um recurso real do sistema. Assim, todos os recursos da rede que devem ser gerenciados são modelados, e as estruturas de dados resultantes são os objetos gerenciados. Os objetos gerenciados podem ter permissões para serem lidos ou alterados, sendo que cada leitura representará o estado real do recurso e, cada alteração também será refletida no próprio recurso [SAMPAIO 97].

Dessa forma, a MIB é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede, possibilitando assim, a automatização de grande parte das tarefas de gerência.

Os padrões de gerenciamento OSI e Internet definiram MIBs que representam os objetos necessários para a gerência de seus recursos.

3.3.3.1. MIB OSI

O padrão OSI define três modelos para gerência de redes: o modelo organizacional, o modelo informacional e o modelo funcional. O modelo organizacional descreve a forma pela qual a gerência pode ser distribuída entre domínios e sistemas dentro de um domínio. O modelo funcional descreve as áreas funcionais e seus relacionamentos. Já o modelo informacional provê a base para a definição de objetos gerenciados e suas relações, classes atributos, ações e nomes.

Na definição de objetos gerenciados é utilizada a orientação a objetos. Objetos com características semelhantes são agrupados em classes de objetos. Uma classe pode ser uma subclasse de outra, e a primeira herda todas as propriedades da segunda. Uma classe é definida pelos atributos da classe, pelas ações que podem ser invocadas, pelos eventos que podem ser relatados, pela subclasse a qual ela deriva e pela superclasse na qual ela está contida.

Para a definição dos objetos gerenciados deve-se considerar três hierarquias: hierarquia de herança, de nomeação e de registros usados na caracterização e identificação de objetos gerenciados.

A seguir são apresentadas cada uma das hierarquias mencionadas acima.

- **Hierarquia de Herança** – Também denominada hierarquia de classe, tem como objetivo facilitar a modelagem dos objetos, através da utilização do paradigma da orientação a objetos. Assim podem ser definidas classes, superclasses, subclasses. Trata-se de uma ferramenta para uma melhor definição de classes.

- **Hierarquia de Nomeação** – A hierarquia de nomeação, também chamada hierarquia de containment, descreve a relação de "estar contido em" aplicado aos objetos. Um objeto gerenciado está contido dentro de um e somente um objeto gerenciado. Um objeto gerenciado existe somente se o objeto que o contém existir, e dependendo da definição, um objeto só pode ser removido se aqueles que lhe pertencerem forem removidos primeiro.

- **Hierarquia de Registro** – A hierarquia de registro é usada para identificar de maneira universal os objetos, independentemente das hierarquias de heranças e nomeação. Esta hierarquia é especificada segundo regras estabelecidas pela notação ASN.1 (Abstract Syntax Notation. 1). Assim, cada objeto é identificado por uma sequência de números, correspondente aos nós percorridos desde a raiz, até o objeto em questão. Esta hierarquia é também usada pelo padrão Internet e será vista com mais detalhes a frente.

3.3.3.2. MIB Internet

O RFC 1066 apresentou a primeira versão da MIB para uso com o protocolo TCP/IP, a MIB-I. Este padrão explicou e definiu a base de informação necessária para monitorar e controlar redes baseadas no protocolo TCP/IP. O RFC 1066 foi aceito pela IAB (Internet Activities Board) como padrão no RFC 1156.

O RFC 1158 propôs uma Segunda MIB, a MIB-II, para uso com o protocolo TCP/IP, sendo aceita e formalizada como padrão no RFC 1213. A MIB-II expandiu a base de informações definidas na MIB-I.

No padrão Internet os objetos gerenciados são definidos em uma árvore de registro, equivalente a hierarquia de registro do padrão OSI, e que será descrita com maiores detalhes a seguir.

A árvore MIB Internet

A MIB Internet usa uma arquitetura de árvore (veja a figura 1), definida na Isso ASN.1, para organizar todas as suas informações. Cada parte da informação da árvore é um nó rotulado que contém:

- um identificador de objetos (OID): seqüência de números separados por pontos;
- uma pequena descrição textual: descrição o nó rotulado .

Exemplo:

directory(1)

identificador de objetos: 1.3.6.1.1

descrição textual: {internet 1}

Um nó rotulado pode ter subárvores contendo outros nós rotulados. Caso não tenha subárvores, ou nós folhas, ele conterá um valor e será um objeto.

O nó raiz da árvore MIB não tem nome ou número, mas tem três sub-árvores:

1. ccitt(0), administrada pelo CCITT;
2. iso(1), administrada pela ISO;
3. joint-isso-ccitt(2), administrada pela Isso juntamente com o CCITT.

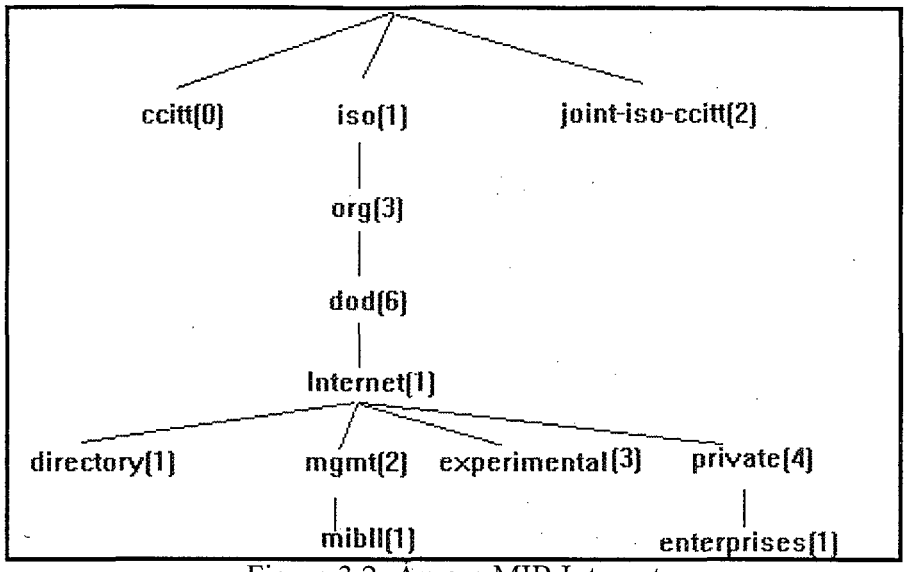

Figura 3.2: Árvore MIB Internet

Sob o nó iso(1), estão outras sub-árvores, como é o caso da sub-árvore org(3), definida pela ISO para conter outras organizações. Uma das organizações que está sob a sub-árvore org(3) é o Departamento de Defesa dos Estados Unidos (DOD), no nó dod(6). A Internet(1) está sob o dod(6), e possui quatro subárvores:

1. directory(1): contém informações sobre o serviço de diretórios OSI (X.500);
2. mgmt(2): contém informações de gerenciamento, é sob esta subárvore que está o nó da mibII, com o identificador de objeto 1.3.6.1.2.1 ou { mgmt 1 };
3. experimental(3): contém os objetos que ainda estão sendo pesquisados pela IAB;
4. private(4): contém objetos definidos por outras organizações.

Abaixo da subárvore mibII estão os objetos usados para obter informações específicas dos dispositivos da rede. Esses objetos são divididos em 11 grupos, que são apresentados na tabela 3.1.

Tabela 3.1: Grupos da MIB-II

Grupos	**Informações**
System(1)	Sistema de operação dos dispositivos da rede
Interfaces(2)	Interface da rede com o meio físico
Address translation(3)	Mapeamento de endereços IP em endereços físicos
Ip(4)	Protocolo IP
Icmp(5)	Protocolo ICMP
Tcp(6)	Protocolo TCP
Udp(7)	Protocolo UDP
Egp(8)	Protocolo EGP
Cmot(9)	Protocolo CMOT
Transmission(10)	Meios de transmissão
Snmp(11)	Protocolo SNMP

Cada objeto contido nos grupos apresentados na tabela 1 é descrito no RFC1213. A descrição dos objetos é dividida em cinco o partes: o nome do objeto, a sintaxe abstrata do objeto, a descrição textual do significado do objeto, o tipo de acesso permitido ao objeto (read-only, read-write, write-only ou não acessível), e o estado do objeto (obrigatório, opcional, obsoleto). O exemplo abaixo apresenta a descrição do objeto sysDescr{Internet 1}.

OBJECT

sysDescr{Internet 1}

Syntax:

DisplayString(SIZE(0..255))

Definition: descrição textual da entidade. Nome completo, vesão, etc.

Access: read-only

Status: mandatory

3.3.3.3. Comparação entre a MIB da OSI e a MIB da Internet

As MIB's da ISO e da Internet são modeladas através de técnicas de programação por objeto. Dentro deste contexto, os recursos a serem gerenciados são representados através de objetos gerenciados.

A diferença entre estas duas MIB's reside nas hierarquias usadas para representar os objetos. Na MIB da ISO são definidas três hierarquias: hierarquia de herança, hierarquia de nomeação e hierarquia de registro.

A hierarquia de herança ou de classes está relacionada às propriedades associadas a um determinado objeto. Dentro desta hierarquia diz-se que objetos da mesma classe possuem propriedades similares.

No caso da Internet não são usados os conceitos de classes de objetos e seus respectivos atributos. São definidos tipos de objetos. A definição de tipo de objetos contém cinco campos: nome textual com o respectivo identificador de objeto (*OBJECT IDENTIFIER*), uma sintaxe ASN.1, uma descrição do objeto, o tipo de acesso e o status.

A hierarquia de nomeação é usada para identificar instâncias de objetos. Este tipo de hierarquia não é definido no caso da Internet.

Finalmente tem-se a hierarquia de registro que é especificada em ambos padrões.

3.3.4. Baseline

A consulta à MIB de um determinado equipamento gerenciável da rede, nos informa uma instância da variável consultada. Porém, para definir quando um segmento monitorado está tendendo a um estado crítico, é preciso comparar a situação atual deste segmento com os dados estatísticos sobre o comportamento considerado normal para este segmento. Daí a necessidade da criação de *baselines*.

A *baseline* é uma caracterização estatística do funcionamento normal do segmento monitorado da rede [NETO 98]. Ela contém as informações sobre o funcionamento normal da rede; podendo ser criadas de diversas formas, as mais comuns são através de técnicas de simulação ou de monitoração da rede [FRANCESCHI 97].

Uma proposta para gerar uma *baseline* monitorada [ROCHA 97] é o monitoramento da máquina *gateway* durante um período determinado, e a partir dos dados colhidos, estimar o comportamento para cada hora de cada dia da semana, gerando vários arquivos, um para cada parâmetro utilizado no sistema especialista de verificação.

Uma das principais funções de um sistema gerente pró-ativo é a notificação do gerente da rede, ou seja, informar o gerente sobre a ocorrência de situações que possam levar à degradação do sistema e sugerir ações pró-ativas a serem tomadas.

Para que seja feita uma boa notificação de previsões, o processo de diagnóstico é adaptado à monitoração constante da rede. Segundo [BOWERMAN 87], a escolha da técnica de previsão deve levar em consideração os seguintes fatores:

- **forma de previsão**: basicamente pode ser feita de duas formas: estimação de parâmetros ou por modelos de regressão;

- **intervalo de tempo**: identifica os períodos de tempo em que se deseja trabalhar, já que as previsões são feitas para intervalos de tempo;

- **padrão dos dados**: identificação de padrões, tendências e sazonalidades existentes nos dados;

- **custo de previsão**: dentre os custos, destacam-se: o custo de desenvolvimento do modelo, o custo do armazenamento das informações necessárias à previsão e o custo do cálculo da previsão;

- **erro amostral**: grau de confiança da previsão;

- **disponibilidade dos dados;**

- **facilidade de operação e compreensão.**

Como pode-se constatar em [FRANCESCHI 97], o processo de diagnóstico pode ser realizado utilizando um sistema especialista. Onde o conhecimento é representado através de fatos e regras. Os fatos são obtidos através dos dados contidos na *baseline*, as regras são determinadas por um especialista. Como poderá ser constatado posteriormente neste trabalho, a criação de *baselines* através de métodos bayesianos utiliza outra estratégia.

A atualização da *baseline* deve ser feita regularmente, devido às rápidas mudanças na área da informática, ou sempre que o segmento monitorado for modificado. Uma *baseline* desatualizada, pode refletir uma rede não correspondente à rede atual, isto implica em notificações e ações tomadas sem necessidade, e outras, necessárias, não notificadas. Ocasionando, desta maneira, o mau gerenciamento da rede.

Até o presente momento as *baselines* foram criadas de forma estática, sem mecanismos de atualização automática. O problema é que depois de algum tempo, estas *baselines* não refletem mais o comportamento da rede.

Neste trabalho é mostrado que uma Rede Bayesiana de Conhecimento pode ser utilizada como *baseline*, apresentando um novo conceito na área de Sistemas Especialistas de Gerência de Redes, o conceito de "*Baselines* Dinâmicas". Dentre as vantagens de seu uso destaca-se que ela atualiza-se com as mudanças da rede e que ela reflete o comportamento da rede através de probabilidades.

CAPÍTULO IV

4. Domínio de Aplicação

Neste trabalho, o domínio de aplicação é a área de gerência de redes.

4.1. Escopo do Trabalho

Existem diversas plataformas de gerência de redes no mercado como por exemplo o *NetView*, da IBM [TIVOLI 99]; a *Sun Net Manager* da SUN[SUN 00] e a *HP OpenView* da Hewlett Packard – HP [HP 00]. Mesmo utilizando as facilidades destas plataformas, como monitoração de variáveis e visualizações gráficas, gerenciar uma rede ainda é uma tarefa difícil. A todo momento o administrador, que tem suas decisões apoiadas basicamente em dados, depara-se com situações que exigem tratamento da incerteza. Apesar dos serviços oferecidos, nenhuma das plataformas atuais é capaz de identificar problema e sugerir ações corretivas, deixando ao administrador o encargo de interpretar gráficos e valores de variáveis.

Em uma rede o fato que merece maior destaque é o congestionamento. Normalmente a situação de congestionamento vai evoluindo aos poucos, e caso o gerente

não tome alguma atitude, pode até levar à paralisação completa da rede. O congestionamento pode ocorrer devido a erros ou até mesmo pela sobrecarga de alguma parte da rede, formando um "gargalo". Para este fato será dada maior evidência no escopo deste trabalho.

4.2. Domínio da Aplicação

As variáveis monitoradas nos dão informações sobre o tráfego existente entre a RNP, Rede Nacional de Pesquisa e a RCT, Rede Catarinense de Ciência e Tecnologia, pois são relativas ao roteador e a respectiva porta que realiza a comunicação entre estas duas redes, conforme a figura 4.1.

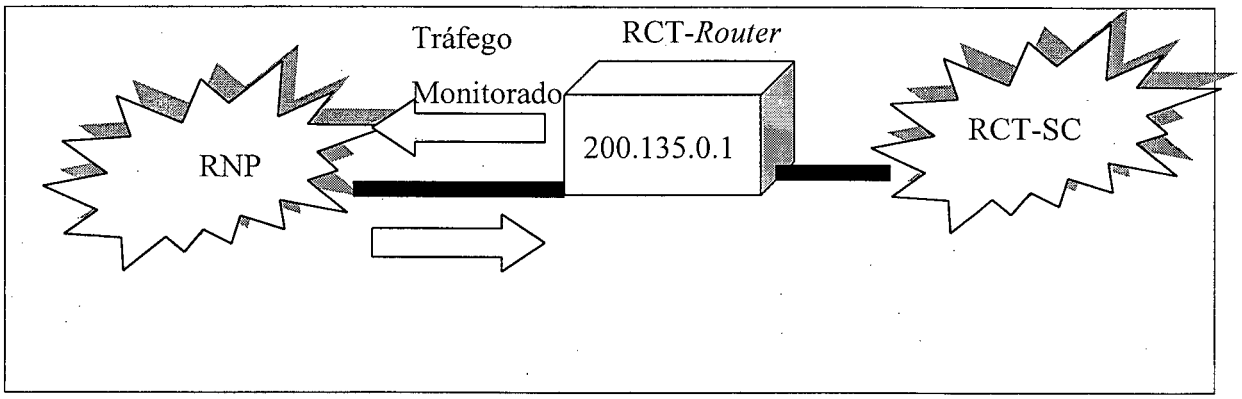

Figura 4.1: Tráfego monitorado

A RNP – Rede Nacional de Pesquisa é um Programa Prioritário do MCT – Ministério da Ciência e Tecnologia, apoiado e executado pelo CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, cuja missão principal é operar um serviço de backbone Internet voltado à comunidade acadêmica e de pesquisa. Atualmente, a RNP conecta 26 dos 27 estados brasileiros, interligando dezenas de milhares de computadores

em todo o país. Diversos centros e institutos de pesquisa e instituições federais de ensino superior fazem uso intensivo da Internet através dos serviços da RNP. A utilização de sua infra-estrutura permite o desenvolvimento concreto de uma rede Internet brasileira, voltada para educação e pesquisa, com tecnologia comparável a dos países mais avançados.[RNP 00]

A Rede de Ciência e Tecnologia de Santa Catarina, RCT-SC, resultou de um convênio entre o Estado de Santa Catarina, através da ex-Secretaria de Estado da Tecnologia, Energia e Meio-Ambiente – SETEMA (hoje representada pelo FUNCITEC), da Secretaria de Educação, Cultura e Desporto – SED, da Universidade do Estado de Santa Catarina – UDESC, da Empresa de Pesquisa e Agropecuária e Difusão de Tecnologia de Santa Catarina – EPAGRI, com a Universidade Federal de Santa Catarina - UFSC, a Federação das Indústrias do Estado de Santa Catarina – FIESC, o Serviço de Apoio à Pequena e Média Empresa – SEBRAE e a Associação Catarinense das Fundações Educacionais – ACAFE. A RCT-SC é uma extensão estadual da Rede Nacional de Pesquisa – RNP e da INTERNET – Rede Internacional de Computadores, e se constitui na infra-estrutura básica do Sistema Estadual de Informação em Ciência e Tecnologia [RCT 00].

4.3. O Roteador Monitorado

O roteador monitorado é da marca Cisco, da série Cisco 7000 e localiza-se fisicamente nas instalações do Núcleo de Processamento de Dados da Universidade Federal de Santa Catarina em Florianópolis, SC. Este roteador é multiprotocolo. As *interfaces* de rede consistem em processadores de *interface* modulares, que oferecem uma conexão direta entre os barramentos de alta velocidade Cisco Extended Bus (CxBus) e uma rede externa [CISCO 99].

4.4. As Variáveis Monitoradas

As variáveis monitoradas pertencem ao grupo *interfaces* da MIB2 – Internet especificada na RFC 1213 [IETF 99]. O Grupo *Interfaces* oferece dados sobre cada *interface* de um dispositivo gerenciável da rede. Essas informações são úteis para o gerenciamento de falhas, de configuração, de desempenho, e de contabilização. As variáveis monitoradas são:

- **ifOutOctets** – o número total de bytes transmitidos por uma *interface*, incluindo caracteres de cabeçalho. Nome: IF-MIB!ifOutOctets; Identificador: 1.3.6.1.2.1.2.2.1.16;

- **ifOutDiscards** – o número de pacotes a serem transmitidos por uma *interface* acima do limite. Estes pacotes são escolhidos para serem descartados mesmo que não tenham sido detectados erros. Nome: IF-MIB!ifOutDiscards; Identificador: 1.3.6.1.2.1.2.2.1.19;

- **ifOutErrors** – o número de unidades de transmissão que contiveram erros. Estes pacotes ou unidades de transmissão são descartados, impedindo que os mesmos se propaguem pela rede. Nome: IF-MIB!ifOutErrors; Identificador: 1.3.6.1.2.1.2.2.1.20;

- **ifInOctets** – o número total de bytes recebidos em uma *interface*, incluindo caracteres de cabeçalho. Nome: IF-MIB!ifInOctets; Identificador: 1.3.6.1.2.1.2.2.1.10;

- **ifInDiscards** – o número de pacotes recebidos em uma *interface* acima do limite. Estes pacotes são escolhidos para serem descartados mesmo que não tenham sido detectados erros. Uma razão para descartar pacotes é que ele pode ser maior do que o espaço livre no *buffer* do roteador. Nome: IF-MIB!ifInDiscards; Identificador: 1.3.6.1.2.1.2.2.1.13;

- **ifInErrors** – o número de unidades de transmissão recebidos com erros. Estes pacotes ou unidades de transmissão são descartados, impedindo que os erros se propaguem para o protocolo de nível mais alto. Nome: IF-MIB!ifInErrors; Identificador: 1.3.6.1.2.1.2.2.1.14;

CAPÍTULO V

5. O Sistema Implementado

As informações que o módulo *baseline* contém sobre o comportamento atual da rede podem ser utilizadas para fazer as estimativas numéricas das probabilidades que evidenciam relações entre os estados da rede e a ocorrência de problemas. Para isto pode ser utilizado técnicas de *Data Mining* [FAYYAD 96], como uma primeira estimativa destas probabilidades. Este trabalho utiliza técnicas estatísticas no processo de *Data Mining*.

Este trabalho explora a aplicação do modelo bayesiano no apoio à gerência de redes. Mais especificamente, o sistema implementado resulta em uma *baseline* bayesiana, que pode ser utilizada para realizar previsões sobre o comportamento de tráfego da rede, podendo ser utilizada também por sistemas especialistas de gerência de redes.

5.1. Arquitetura do Sistema

A arquitetura do Sistema de Gerência de Redes Bayesiano (SISGEBAY) implementado neste trabalho pode ser vista na figura 5.1, e é composta das seguintes partes:

- **Estação gerente** - Estação de trabalho onde é executado o programa gerente, responsável pela coleta de dados do roteador;

- **Roteador** - roteador monitorado;

- **MIB II** - MIB Internet a qual as variáveis coletadas pertencem;

- **Agente** – Feita a solicitação de coleta dos dados pela estação gerente, o agente é responsável por coletar os dados da MIB II e gravá-los em um arquivo;

- **Base de dados** - Arquivos onde serão gravados os dados coletados do roteador;

- **KDD** – Processo responsável pela preparação dos dados coletados e pelo processo de *Data Mining*. O processo de KDD fará uso da *shell* Netica, para o *Data Mining*.

- ***Shell-*** Armazena a base de conhecimento probabilístico e permitirá a visualização da Rede Bayesiana que representará o conhecimento incerto. Após a definição das variáveis e regras e/ou fatos, a *shell* gera uma distribuicão conjunta de probabilidades;

- ***Baseline* bayesiana** – É composta por variáveis, seus respectivos atributos e, pela relação entre as variáveis que é estabelecida através da definicão de regras e/ou fatos. Esta *baseline* é representada por distribuições de probabilidades para as hipóteses diagnósticas.

- **Interface** – Tem como objetivo principal, fazer a comunicação entre o usuário e a *baseline* bayesiana, exibe todas as informações durante as consultas.

- **Usuário** – Estação que fará uso das informações oferecidas pelo SISGEBAY.

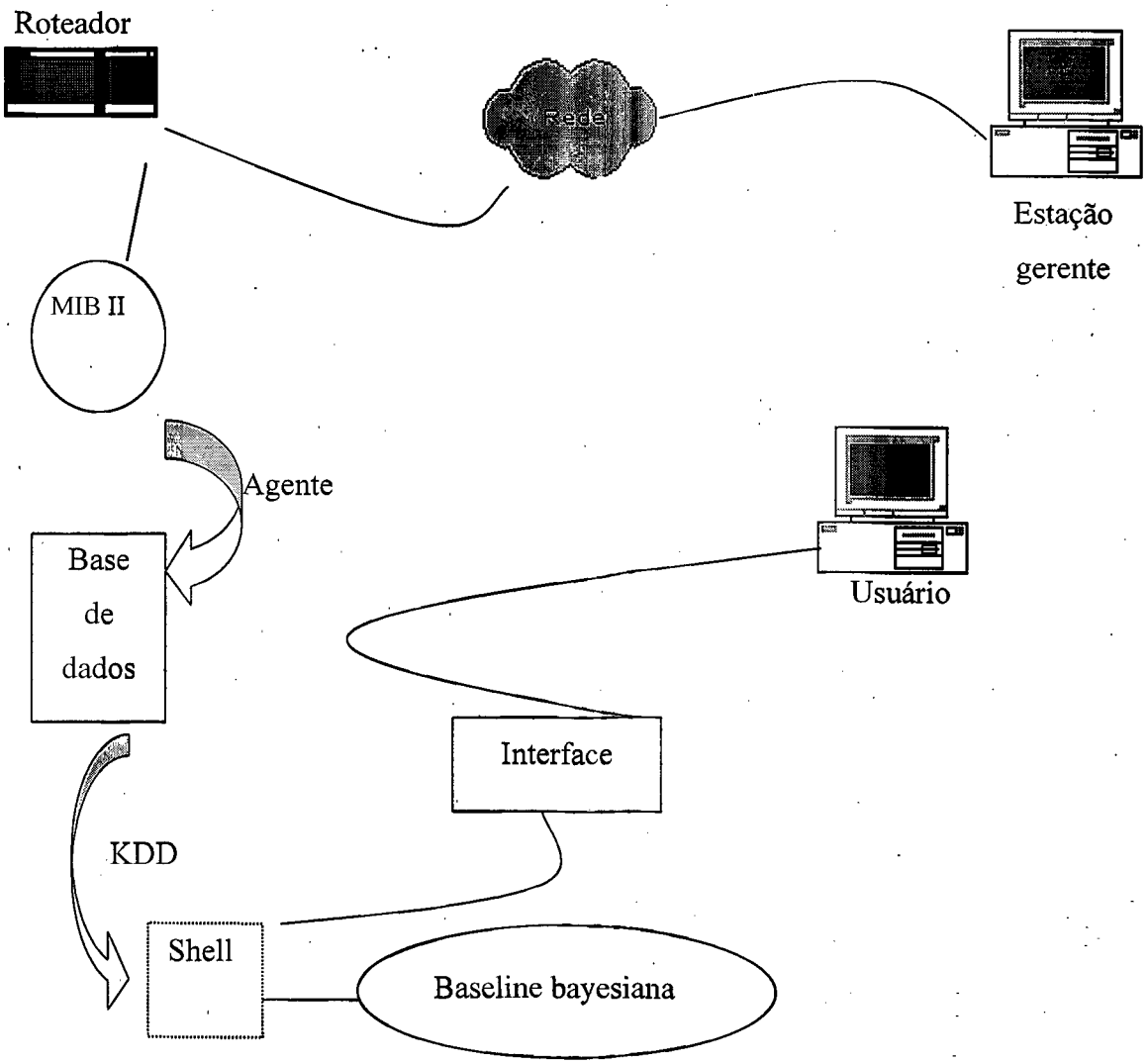

Figura 5.1: Arquitetura do Sistema

5.2. Metodologia de Desenvolvimento

Este trabalho consiste em primeiramente monitorar um segmento da rede, coletando e gravando os dados. Paralelamente à coleta dos dados é criada uma base de dados com os valores das variáveis monitoradas. Técnicas estatísticas de *Data Mining* são aplicadas na base de dados para criar a *baseline* bayesiana. Este trabalho culmina com a

implementação de um sistema que, utilizando a *baseline* bayesiana, é capaz de predizer o comportamento do segmento monitorado da rede.

Este trabalho foi desenvolvido em duas etapas: a primeira etapa consistiu em um estudo preliminar, onde várias tarefas foram realizadas manualmente; a segunda etapa deste trabalho consistiu na automatização de tarefas que eram realizadas manualmente na primeira etapa. Erros de coleta e variáveis coletadas não significativas foram tratadas na segunda etapa, com base nos estudos realizados na primeira etapa.

5.3. Java

O Java [DACONTA 96] é uma linguagem de programação cuja utilização avança exponencialmente. Sua simplicidade e independência de plataforma, viabiliza a implementação deste trabalho e confere ao mesmo inúmeras capacidades de expansões futuras.

Ao ser compilado um programa em Java, é gerado um código para uma máquina genérica denominado *byte code*. Este código pode ser executado tanto por interpretadores como por *Browsers Web* (que simulam a implementação de uma máquina virtual baseada em *byte code*). Devido a esta característica os programas em Java são independentes de plataforma.

Com o Java é possível a criação de dois tipos de programas: *applets* e *applications*. Os *Applets* são aplicações desenvolvidas em Java destinadas a serem executadas em *Browsers Web*. Uma *application* é uma aplicação destinada a ser executada a partir de um interpretador.

O Java, por ser uma linguagem baseada em objetos, permite a construção de programas legíveis e de fácil manutenção [BERNHARDT 96]. Estas qualidades, associadas à característica dinâmica de sua *interface*, permitem que seus programas sejam muito mais fáceis de serem modificados do que um escrito em linguagem baseada em *script*.

5.4. A coleta e preparação dos dados na primeira etapa do trabalho

Foram coletados dados relativos à *interface* serial 4 do roteador citado anteriormente, utilizando para isto um programa escrito em Java. Os dados foram coletados periodicamente em espaços de aproximadamente cinco minutos. Para este trabalho foram selecionados os dados coletados do dia 24 de dezembro de 1998 ao dia 24 de janeiro de 1999.

5.4.1. O programa de coleta dos dados

Foi criada uma *application* em Java para realizar a coleta dos dados. Esta *application* ficou sendo executada em uma estação de trabalho IBM RS/6000, coletando os valores das variáveis monitoradas e os valores de suas taxas médias.

Foram utilizadas constantes para facilitar a visualização e entendimento do programa. Veja o quadro 5.1.

Quadro 5.1: Constantes definidas no programa de coleta de dados

```
Static String roteador = "200.135.0.1";
static String ifOutOctets = ".1.3.6.1.2.1.2.2.1.16.4";
static String ifOutDiscards = ".1.3.6.1.2.1.2.2.1.19.4";
static String ifOutErrors = ".1.3.6.1.2.1.2.2.1.20.4";
static String ifInOctets = ".1.3.6.1.2.1.2.2.1.10.4";
static String ifInDiscards = ".1.3.6.1.2.1.2.2.1.13.4";
static String ifInErrors = ".1.3.6.1.2.1.2.2.1.14.4";
```

Além de utilizar o identificador da variável monitorada é necessário colocar, no final do identificador, o número da porta monitorada. Por exemplo: a constante ifOutOctets

tem como identificador o número 1.3.6.1.2.1.2.2.1.16 porém para a coleta de dados, deve-se adicionar o ".4" indicando que a porta monitorada é a porta 4 [VERONEZ 99].

Os dados coletados são gravados em um arquivo texto, no formato apresentado no quadro 5.2.

Quadro 5.2: Exemplo do arquivo de dados

```
Mon Jan 18 16:15:14 GMT-02:00 1999 315 8.05594119E8  802270 254.0  0 0.0
0 2.599544099E9  912464 0.0  0 4378.0  0
```

Os primeiros valores, correspondem a data, o próximo valor (no caso da primeira linha o valor 315) corresponde ao tempo decorrido em segundos. Veja que o valor do tempo decorrido varia de uma coleta para outra, esta variação é decorrente do tempo de rede existente entre a comunicação do processo gerente (esta *application*) e o agente no roteador. Por trabalhar-se com taxas médias esta variação é irrelevante para a aplicação final.

Os próximos valores correspondem respectivamente ao valor e a taxa média das variáveis: ifOutOctets; ifOutDiscards; ifOutErrors; ifInOctets; ifInDiscards; ifInErrors. O programa grava todas as taxas em bits por segundo (bps).

Como as variáveis monitoradas são contadores, elas podem retornar a zero, ocasionando erros nas taxas. Outro erro que podemos encontrar é devido a falhas de comunicação na rede. No caso de ocorrência de erros, no arquivo de dados é gravado "erro" e no arquivo de erros é gravado a data, horário, tipo do erro e onde o erro ocorreu, conforme o quadro 5.3.

Quadro 5.3: Exemplo do arquivo de erros

```
Fri Jan 22 18:37:02 GMT-02:00 1999 0 Tempo decorrido negativo ou igual a
zero
Sat Jan 23 12:33:50 GMT-02:00 1999 timed out na comunicacao para:
200.135.0.30 Erro no ifOutOctets
```

Ao final de uma coleta o programa "aguarda" por um período de aproximadamente cinco minutos, até ser novamente executado. A coleta dos dados relativos às outras cinco variáveis é semelhante à coleta do ifOutOctets apresentada no quadro 5.4.

Quadro 5.4: Coleta dos dados do ifOutOctets

```
//Inicio do ifOutOctets
   long ifOutOctetsAnterior=ifOutOctetsAtual;
   ResultString r=snmp.retornaValor(roteador,ifOutOctets);
   if (r.erro)
     {
     fileerro.println(now + " " + r.retornaMensagemDeErro() +
     " Erro no ifOutOctets ");
     file.print("Erro ");
     }
   else
     {
     Double ifOutOctetsStr=new Double (r.retornaResultado());
     //System.out.println("ifOutOctets: "+ ifOutOctetsStr);
     file.print (ifOutOctetsStr+"  ");
     ifOutOctetsAtual=ifOutOctetsStr.longValue();
     long ifOutOctetsTaxa=(((ifOutOctetsAtual-
ifOutOctetsAnterior)*8)/TempoDecorrido);
     file.print(new Long(ifOutOctetsTaxa).toString()+" ");
     }
//Fim do ifOutOctets
```

5.4.2. A preparação dos dados

Para a preparação dos dados foi utilizada a planilha eletrônica Excel [MICROSOFT 99].

Algumas colunas dos dados coletados não foram utilizadas neste trabalho, por isso foram eliminadas, mantendo-se apenas as colunas relativas ao dia da semana, ao dia, ao horário e às taxas médias das seis variáveis monitoradas.

As linhas onde ocorreram erros na coleta também foram excluídas, a manutenção destas linhas poderia alterar a conclusão do processo de KDD e sua exclusão não afeta o resultado final, pois trabalhou-se com os valores das taxas médias das variáveis.

Os dados coletados foram agrupados de hora em hora, com suas respectivas taxas médias. Os dias da semana também foram agrupados com suas médias. Desta forma a base de dados foi transformada para explicitar o comportamento médio das variáveis coletadas de acordo com a faixa horária e o dia da semana. A tabela 5.1 mostra parte da base de dados preparada.

Tabela 5.1: Base de dados preparada

Dia	Horário	Taxa Média ifOutOctets	Taxa Média ifOutDiscards	Taxa Média ifOutErrors	Taxa Média ifInOctets	Taxa Média ifOutDiscards	Taxa Média ifOInErrors
Sex	22:00	386758,454	0	0	343608,363	0	0
Sex	23:00	302198,181	0	0	352445,545	0	0
Sáb	00:00	318300,417	0	0	365418,083	0	0
Sáb	01:00	366227,181	0	0	406386,545	0	0

A partir desta base de dados, foram encontradas as probabilidades *a priori* **P(Hi)** e as probabilidades condicionais **P(e|Hi)**, conforme as tabelas 5.2, 5.3, 5.4, 5.5 e 5.6.

Tabela 5.2: Probabilidades das hipóteses diagnósticas

Hipóteses Diagnósticas	**P(Hi)**
Intenso	0,090
Normal	0,530
Fraco	0,370
Muito_fraco	0,010

Tabela 5.3: Probabilidades condicionais da evidência Dia da Semana

Dia da Semana	**P(ek|Intenso)**	**P(ek|Normal)**	**P(ek|Fraco)**	**P(ek|Muito_fraco)**
Segunda	0,200	0,180	0,100	0,050
Terça	0,200	0,180	0,110	0,050
Quarta	0,170	0,180	0,140	0,050
Quinta	0,160	0,190	0,110	0,050
Sexta	0,150	0,150	0,140	0,050
Sábado	0,100	0,090	0,200	0,300
Domingo	0,020	0,030	0,200	0,450

Tabela 5.4: Probabilidades condicionais da evidência Taxa Média ifInOctets

Taxa Média ifInOctets	**P(ek|Intenso)**	**P(ek|Normal)**	**P(ek|Fraco)**	**P(ek|Muito_fraco)**
Até 240 Kbps	0,000	0,000	0,950	0,100
De 240 a 410 Kbps	0,000	0,730	0,050	0,000
De 410 a 580 Kbps	0,000	0,270	0,000	0,000
Acima de 580 Kbps	1,000	0,000	0,000	0,000

Tabela 5.5: Probabilidades condicionais da evidência Taxa Média ifOutOctets

Taxa Média ifOutOctets	**P(ek|Intenso)**	**P(ek|Normal)**	**P(ek|Fraco)**	**P(ek|Muito_fraco)**
Até 240 Kbps	0,000	0,000	0,810	1,000
De 240 a 410 Kbps	0,000	0,550	0,190	0,000
De 410 a 580 Kbps	0,000	0,320	0,000	0,000
Acima de 580 Kbps	1,000	0,130	0,000	0,000

Tabela 5.6: Probabilidades condicionais da evidência Horário

Horário	P(ek|Intenso)	P(ek|Normal)	P(ek|Fraco)	P(ek|Muito_fraco)
Zero	0,020	0,040	0,050	0,060
Uma	0,020	0,040	0,050	0,060
Duas	0,020	0,040	0,050	0,060
Três	0,020	0,040	0,050	0,060
Quatro	0,020	0,040	0,050	0,060
Cinco	0,020	0,040	0,050	0,060
Seis	0,020	0,040	0,050	0,060
Sete	0,070	0,040	0,030	0,010
Oito	0,070	0,050	0,030	0,020
Nove	0,070	0,050	0,030	0,020
Dez	0,070	0,040	0,030	0,020
Onze	0,070	0,040	0,030	0,010
Doze	0,070	0,040	0,030	0,010
Treze	0,070	0,040	0,030	0,010
Quatorze	0,090	0,050	0,030	0,020
Quinze	0,070	0,050	0,0300	0,020
Dezesseis	0,070	0,040	0,0300	0,020
Dezessete	0,020	0,040	0,0500	0,060
Dezoito	0,020	0,040	0,0500	0,060
Dezenove	0,020	0,040	0,0500	0,060
Vinte	0,020	0,040	0,0500	0,060
Vinte e uma	0,020	0,040	0,0500	0,060
Vinte e duas	0,020	0,040	0,0500	0,060
Vinte e três	0,020	0,040	0,0500	0,060

As demais evidências foram desconsideradas na rede bayesiana, pois, pelos dados coletados, elas se mantém constantes (e seus valores próximos de zero), portanto a certeza da ocorrência destas evidências não afeta os valores das probabilidades das hipóteses diagnósticas[VERONEZ 99].

5.5. A coleta e preparação dos dados na segunda etapa do trabalho

Durante esta etapa foram coletados dados relativos à mesma *interface* e roteador da primeira etapa, ou seja, dados relativos à *inteface* serial 4 do roteador 200.135.0.1, citado anteriormente, utilizando para isto um programa escrito em Java. Os dados foram coletados periodicamente em espaços de aproximadamente cinco minutos entre cada coleta.

Para facilitar o entendimento e evitar redundâncias, nos exemplos apresentados são utilizados os mesmos valores de dados nas duas etapas.

5.5.1. O programa de coleta dos dados

Foi criada uma *application* em Java para realizar a coleta dos dados, através da remodelação da *application* de coleta apresentada anteriormente. Esta *application* ficou foi executada em um micro IBM/PC, coletando os valores das variáveis monitoradas e os valores de suas taxas médias.

Como poderá ser observado esta nova *application* melhora o desempenho e facilita a automatização do sistema alvo desta pesquisa.

A primeira *application* não possuia interface de controle, nesta segunda etapa foi criada uma interface de controle para a coleta dos dados que pode ser vista na figura 5.2.

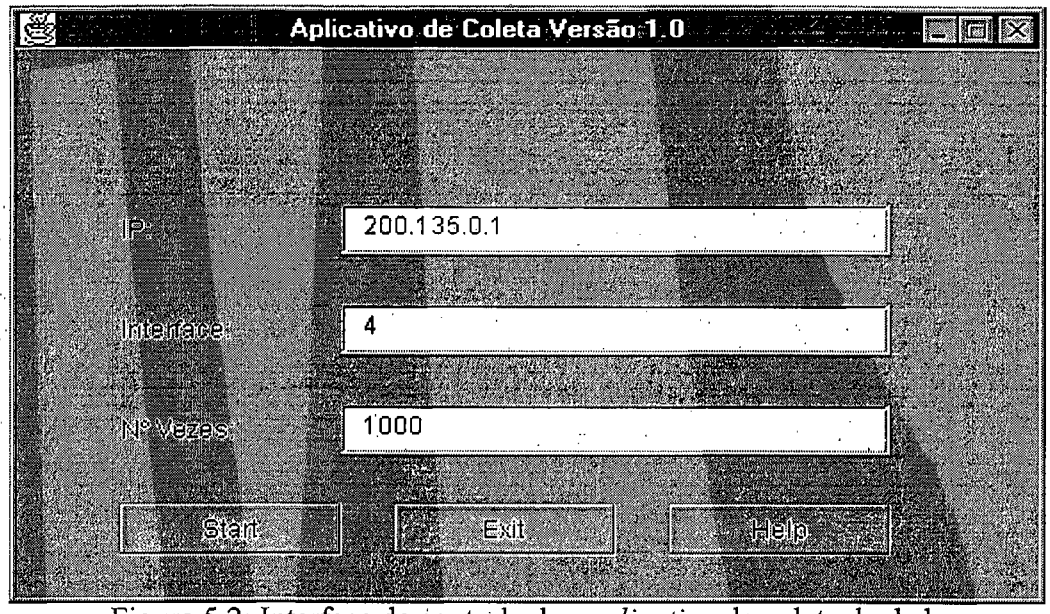

Figura 5.2: Interface de controle da *application* de coleta de dados

O programa de coleta anterior utilizava um arquivo texto para guardar os dados, este programa armazena os dados em um arquivo .MBD. Para a mesma coleta apresentada no quadro 5.2 do programa anterior, o programa atual armazenaria os dados conforme a tabela 5.7.

Tabela 5.7: Exemplo do arquivo de dados do programa de coleta segunda versão

Dia	Mês	Ano	Dia Semana	Hora	Minuto	ifOutOctets	ifInOctets
18	Jan	1999	Mon	16	15	802270	912464

No decorrer do trabalho foi visto que as variáveis ifInErrors, ifOutErrors, ifInDiscards e ifOutDiscards não afetam os valores das probabilidades diagnósticas finais, portanto não são coletadas, bem como suas taxas médias [VERONEZ 99].

O arquivo de erros permanece sendo armazenado conforme o do programa de coleta anterior.

A PDU enviada ao agente solicitando o envio dos valores das variáveis coletadas é montada pelo próprio programa de coleta, dando mais liberdade para qualquer alteração futura. Conforme pode ser visto no quadro 5.5.

Quadro 5.5: Procedimento que monta a PDU e coleta os dados

```
Public String getID (String Xid, String sq) {

      SnmpAPI api;
      String val = "";
      //rfc da mib2
      String mib="Rfc1213-mib";
      api = new SnmpAPI();
      api.start();
      SnmpSession session = new SnmpSession(api);
      try {
         //System.out.println("thread dormindo...");
         Thread.sleep(10);
         System.out.println("thread acordou...");
          } catch (Exception ie) {
           //System.out.println("Erro: " + ie);
          }
      session = new SnmpSession(api);
      session = new SnmpSession(api);
      session.peername = sq;
      session.community = "public";
      SnmpPDU pdu = new SnmpPDU(api);
      pdu.command = api.GET_REQ_MSG;
      //System.out.println("pduCommand: " + pdu.command);
      SnmpOID oid;
      oid = new SnmpOID(Xid,api);
      pdu.addNull(oid);

      //System.out.println("Antes da sessao.");
      try {
           session.open();
          //System.out.println("Depois da sessao.");
          //System.out.println("Xid: " + Xid);
          //System.out.println("sq: " + sq);
```

(Continua)

Quadro 5.5: Procedimento que monta a PDU e coleta os dados

(Continuação)

```
            pdu = session.syncSend(pdu);
            //System.out.println("Xid: " + Xid);
            //System.out.println("PDU: " + pdu);
            SnmpVarBind varaux= (SnmpVarBind) pdu.variables.firstElement();
            val=varaux.variable.toString();

            } catch (SnmpException e) {
                 val = "Erro";
                 fileErro.println(tempoAtual + "SnmpException:  "+ e);

              } catch (RuntimeException e) {
                  val = "Erro";
                  fileErro.println(tempoAtual + "RuntimeException:  "+ e);

                } catch (Exception e) {

                   val = "Erro";
                   fileErro.println(tempoAtual + "Exception:  "+ e);
                }
        if (api.client == null) {
                   val = "Erro";
        } // fim do If api.client

      session.close();
      api.stop();
      System.out.println("taxa: " + val);
      return(val);

} // fim do método getID
```

Nesta segunda versão do programa de coleta de dados também é feito um maior controle de erros, procurando obter o máximo de eficiência e confiabilidade do programa, eliminando assim grande parte do trabalho realizado na etapa de preparação dos dados. Veja no Quadro 5.6 parte do procedimento de coleta de dados.

Quadro 5.6: Coleta dos dados

```
      //inicio do laco das coletas
      while (nq != 0){
           nq = nq - 1;

         //Inicio da data, tempo


           if (flag_erro != true) {
              tempoAnterior =tempoAtual;
```

(Continua)

Quadro 5.6: Coleta dos dados

(Continuação)

```
            public tAnterior = tempoAnterior.getTimeInMillis();
        } else { nq++;  }
        tempoAtual = Calendar.getInstance();
        tAtual = tempoAtual.getTimeInMillis(); //get(Calendar.SECOND);
        tempoDecorrido = ((tAtual - tAnterior) / 1000); //divide por
1000

        if (tempoDecorrido <=0) {
            fileErro.println(Long.toString(tAtual) + " " +
Long.toString(tempoDecorrido)+ " Tempo decorrido negativo ou igual a
zero");
             flag_tempo_erro=true;
            System.out.println("Tempo decoprrido menor que 0." +
tempoDecorrido);
        } else {
             flag_tempo_erro=false;
          }

        try {

          //inicio da coleta das variaveis

        flag_erro=false;

        //Inicio do ifOutOctets
        System.out.println(" Inicio do IfInOct...");
         if ((flag_erro!=true) && (!flag_tempo_erro)) {
           System.out.println(" IFOutOctAtual: " + ifOutOctetsAtual);
             ifOutOctetsAnterior = ifOutOctetsAtual;
             System.out.println(" IFIOutOctAnt: " +
ifOutOctetsAnterior);

            ifOutOctetsStr=this.getID(ifOutOctets+interf,nip);
            System.out.println(" Voltou do getID: " + ifOutOctetsStr);

            if (ifOutOctetsStr.equals("Erro")) {
                 fileErro.println(tAtual + " " + " Erro no ifOutOctets
");
                 flag_erro=true;
                 nq++;
              } else {
                 ifOutOctetsAtual =
(Long.decode(ifOutOctetsStr)).longValue();
                 ifOutOctetsTaxa = (((ifOutOctetsAtual-
ifOutOctetsAnterior) * 8) / tempoDecorrido);
                 ifOutOctetsStr = Long.toString(ifOutOctetsTaxa);
              }
          } //Fim do ifOutOctets

        //Inicio do ifInOctets
        System.out.println(" Inicio do IfInOct...");
         if ((flag_erro!=true) && (!flag_tempo_erro)) {
             ifInOctetsAnterior=ifInOctetsAtual;
```

(Continua)

Quadro 5.6: Coleta dos dados

(Continuação)

```
                ifInOctetsStr=this.getID(ifInOctets+interf,nip);
                System.out.print(".");

             if (ifInOctetsStr.equals("Erro")) {
                  fileErro.println(tempoAtual + " " + " Erro no ifInOctets
");
                  flag_erro=true;
                  nq++;
                } else {
                  ifInOctetsAtual =
(Long.decode(ifInOctetsStr)).longValue();
                  ifInOctetsTaxa = (((ifInOctetsAtual -
ifInOctetsAnterior) * 8) / tempoDecorrido);
                  ifInOctetsStr = Long.toString(ifInOctetsTaxa);
                }
             } //Fim do ifInOctets

           } catch (Exception e) {
          System.out.println("Erro nas coletas: " + e);
           }
```

Como pode ser observado no procedimento apresentado no quadro 5.7, abaixo, os dados são gravados somente quando toda a coleta ocorreu com sucesso, evitando que se tenham linhas ou colunas com erros.

Quadro 5.7: Gravação dos dados

```
          //Inicio da gravacao no arquivo de dados
           if ((flag_erro != true) && (flag_tempo_erro != true)){

            dataSistema = Calendar.getInstance();

            try {

                  dia = dataSistema.get(Calendar.DAY_OF_WEEK);
                  diaStr = Integer.toString(dia); // converte dia para
string...

                  mes = dataSistema.get(Calendar.MONTH);
                  mesStr = Integer.toString(mes); // converte mes para
string.. .

                  ano = dataSistema.get(Calendar.YEAR);
                  anoStr = Integer.toString(ano); // converte ano para
string...
```

(Continua)

Quadro 5.7: Gravação dos dados

(Continuação)

```
        sem = dataSistema.get(Calendar.DAY_OF_WEEK_IN_MONTH);
        semStr = Integer.toString(sem);

        horaSistema = Calendar.getInstance();

                hORA = horaSistema.get(Calendar.HOUR);
                hora = Integer.toString(hORA); // converte as hora para
String

                min = horaSistema.get(Calendar.MINUTE);
                minuto = Integer.toString(min); // converte os min. para
String

                Connection con =
DriverManager.getConnection("jdbc:odbc:Coleta", "", "");
               Statement stmt = con.createStatement();
               String sql = "SELECT * FROM Dados";
                 ResultSet rs = stmt.executeQuery(sql);

                 sql = "INSERT INTO TabLogUsuario VALUES
('"+diaStr+"','"+mesStr+"','"+anoStr+"','"+semStr+"','"+hora+"','"+minuto
+"','"+ifOutOctetsStr+"','"+ifInOctetsStr+"')";
               System.out.println ("Gravou!!!");
        stmt.executeUpdate(sql);

                rs.close();
        stmt.close();
               con.close();
         } catch (SQLException e) {
             System.out.println ("Erro!"+e);
         } // fim do try/catch

       } //Fim da gravacao no arquivo de dados
```

5.5.2. A preparação dos dados

A segunda versão do programa de coleta de dados automatiza grande parte da preparação dos dados, por exemplo, a exclusão de linhas e colunas não é necessária uma vez que esta versão do programa grava somente os registros coletados com sucesso. Além disto as variáveis coletadas com o primeiro programa que não foram utilizadas na primeira

versão dos sistema, não são coletadas com o segundo programa, poupando assim tempo de processamento e espaço de armazenamento.

Todo o trabalho que na primeira etapa foi realizado com o Excel, nesta segunda etapa é realizada automaticamente através de uma *application* também desenvolvida em Java.

Esta *application* possui uma interface que possibilita ao usuário definir quais as faixas de valores ele considera o tráfego intenso, normal, fraco e muito fraco, como pode ser observada na figura 5.3.

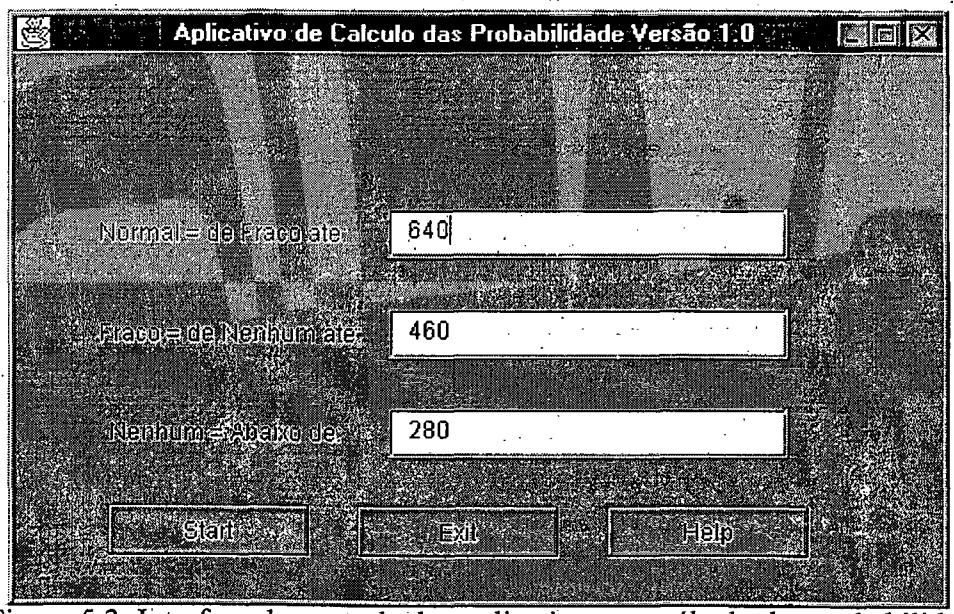

Figura 5.3: Interface de controle da application para cálculo das probabilidades

Esta *application* faz uso de tabelas temporárias para armazenar e contar a ocorrência dos eventos gravados no arquivo de dados. Veja no quadro 5.8 a inicialização das tabelas utilizadas.

Quadro 5.8: Código fonte da inicialização das tabelas

```
/*********** Inicializacao das tabelas ************/

        Object[][] trafegoTable = { {"", "", "", ""} };
        trafegoTable = new Object[0][3];

        for (int y=0; y==3; y++) {
               trafegoTable[0][y] = "0";
        }

        Object[][] diaSemanaTable = { {"","", "", ""} };
        diaSemanaTable = new Object[6][3];

        for (int x=0; x==6; x++) {
        for (int y=0; y==3;y++) {
                     diaSemanaTable[x][y] = "0";
        }
        }

        Object[][] ifInOctetsTable = { {"", "", "", ""} };
        ifInOctetsTable = new Object[3][3];

        for (int x=0; x==3; x++) {
        for (int y=0; y==3;y++) {
                     ifInOctetsTable[x][y] = "0";
        }
        }

        Object[][] ifOutOctetsTable = { {"", "", "", ""} };
        ifOutOctetsTable = new Object[3][3];

        for (int x=0; x==3; x++) {
        for (int y=0; y==3;y++) {
                     ifOutOctetsTable[x][y] = "0";
        }
        }

        Object[][] horarioTable = { {"", "", "", ""} };
        horarioTable = new Object[23][3];

        for (int x=0; x==23; x++) {
        for (int y=0; y==3;y++) {
                     horarioTable[x][y] = "0";
        }
        }
```

Após a criação e inicialização das tabelas temporárias, o arquivo de dados é lido exaustivamente até que sejam computadas todas as coletas armazenadas. A cada registro lido é executado o procedimento de contagem da ocorrência dos eventos, conforme o quadro 5.9.

Quadro 5.9: Contagem da ocorrência dos eventos

```
        /*************** inicio da contagem da ocorrencia dos eventos
*******/

                if              (((((ifOutOctets).longValue())              +
((ifInOctets).longValue()))  / 2) > (normal).longValue()) {

                        coluna = 0; // trafego intenso

                } else { // else do trafego intenso

                        if           ((((ifOutOctets).longValue()           +
(ifInOctets).longValue())/2) > (fraco).longValue()) {

                                coluna = 1; //  trafego normal

                        } else //else do trafego normal

                                if       ((((ifOutOctets).longValue()       +
(ifInOctets).longValue())/2) > (nenhum).longValue()) {

                                        coluna = 2; // trafego fraco
                                } else //else do trafego fraco

                                        coluna = 3; // trafego muito_fraco

                                } //fim do trafego muito_fraco
                        }
                } // fim do if trafego

                trafegoTable[0][coluna] = trafegoTable[0][coluna] + 1;

                swith (diaSemana) {

                        case 1:  //Segunda feira
                                                diaSemanaTable[0][coluna]    =
diaSemanaTable[0][coluna] +1;
                         break;

                        case 2:  //Terca feira
                                                 diaSemnaTable[1][coluna]    =
diaSemanaTable[1][coluna] +1;
                         break;

                        case 3:  //Quarta feira
                                                diaSemanaTable[2][coluna]    =
diaSemanaTable[2][coluna] +1;
                         break;

                        case 4:  //Quinta feira
                                                diaSemanaTable[3][coluna]    =
diaSemanaTable[3][coluna] +1;
                         break;

                        case 5:  //Sexta feira
```

(continua)

Quadro 5.9: Contagem da ocorrência dos eventos

(Continuação)

```
                    diaSemanaTable[4][coluna]                                =
diaSemanaTable[4][coluna] +1;
                     break;

                    case 6:  //Sabado
                                                diaSemanaTable[5][coluna]    =
diaSemanaTable[5][coluna] +1;
                     break;

                    case 7:  //Domingo
                    diaSemanaTable[6][coluna]                                =
diaSemanaTable[6][coluna] +1;
                    break;
             } //fim do Switch


             if (ifInOctets > normal) {

                    ifInOctetsTable[0][coluna]                               =
ifInOctetsTable[0][coluna] + 1;
             } else {
                    if (ifInOctets > fraco) {
                           ifInOctetsTable[1][coluna]                        =
ifInOctetsTable[1][coluna] + 1;
                    } else {
                           if (ifInOctets > muito_fraco) {
                                                ifInOctetsTable[2][coluna]   =
ifInOctetsTable[2][coluna] + 1;
                           } else {
                                  ifInOctetsTable[3][coluna]                 =
ifInOctetsTable[3][coluna] + 1;

                           }
                    }
             }

             if (ifOutOctets > normal) {

                    ifOutOctetsTable[0][coluna]                              =
ifOutOctetsTable[0][coluna] + 1;
             } else {
                    if (ifInOctets > fraco) {
                           ifOutOctetsTable[1][coluna]                       =
ifOutOctetsTable[1][coluna] + 1;
                    } else {
                           if (ifInOctets > muito_fraco) {
                                                ifOutOctetsTable[2][coluna]  =
ifOutOctetsTable[2][coluna] + 1;
                           } else {
                                  ifOutOctetsTable[3][coluna]                =
ifOutOctetsTable[3][coluna] + 1;
```

(Continua)

Quadro 5.9: Contagem da ocorrência dos eventos

(Continuação)

```
                                }
                        }
                }

                for (i=0; i<24; i++) {
                        if (horario == i) {
                                horarioTable[horario][coluna]                =
horarioTable[horario][coluna] +1;
                        }
                }

                } // fim do While


/************** fim da contagem da ocorrencia dos eventos *******/
```

A *application* de preparação dos dados, faz toda a contagem das frequências através da leitura do arquivo gravado na coleta dos dados e grava as probabilidades finais em um outro arquivo .MBD. O arquivo .MBD resultante da aplicação deste programa contém as tabelas 5.2, 5.3, 5.4, 5.5 e 5.6, descritas na primeira etapa de preparação dos dados (seção 5.1), e seus respectivos valores de probabilidades.

5.6. A Shell Utilizada

As *Shells* são *softwares* que facilitam a construção de Sistemas Especialistas pelo fornecimento de esquemas de representação do conhecimento e de máquinas de inferência.

A *Shell* utilizada neste trabalho chama-se Netica [NETICA 00], foi desenvolvida pela *Norsys Software Corp.* em Vancouver, BC, Canadá e utiliza redes de probabilidade para realizar vários tipos de inferência usando algoritmos modernos e rápidos. Dado um novo caso, pelo qual o usuário tem conhecimento limitado, Netica encontrará os valores ou probabilidades apropriadas para todas as variáveis desconhecidas.

Dentre as vantagens da utilização desta *Shell* destacam-se que: pode encontrar decisões ótimas para problemas de decisão sequencial; pode aprender relações probabilísticas através de dados; permite atualização fácil da rede de crença e dos diagramas de influência, incluindo: excluir, colar e duplicar nós da rede de crença e dos diagramas de influência; mantém diagramas complexos ordenados; permite a entrada de relações probabilísticas através de equações, e tem facilidade para realizar a discretização de variáveis contínuas.

5.7. A Rede Bayesiana

5.7.1. A Rede Bayesiana a Priori

Utilizando a *Shell* Netica, dada as probabilidades das Hipóteses e as probabilidades das evidências, foi montada a rede bayesiana *a priori.*

Os cálculos das probabilidades foram realizados de acordo com a Probabilidade Bayesiana. Por exemplo, a probabilidade de ter tráfego na Segunda-feira, **P(Seg)**, é dada por:

$$P(Seg) = P(Intenso \cap Seg) + P(Normal \cap Seg) + P(Fraco \cap Seg) + P(Muito\_fraco \cap Seg)$$

$$P(Seg) = P(Intenso) \cdot P(Seg|Intenso) + P(Normal) \cdot P(Seg|Normal) + P(Fraco) \cdot P(Seg|Fraco) + P(Muito\_fraco) \cdot P(Seg|Muito\_fraco)$$

$$P(Seg) = 0{,}0900 \cdot 0{,}2000 + 0{,}5300 \cdot 0{,}1800 + 0{,}3700 \cdot 0{,}1000 + 0{,}0100 \cdot 0{,}0500 = 0{,}1509$$

A probabilidade de ter tráfego no Sábado, **P(Sab)**, é dada por:

$$P(Sab) = P(Intenso \cap Sab) + P(Normal \cap Sab) + P(Fraco \cap Sab) + P(Muito\_fraco \cap Sab)$$

$$P(Sab) = P(Intenso) \cdot P(Sab|Intenso) + P(Normal) \cdot P(Sab|Normal) + P(Fraco) \cdot P(Sab|Fraco) + P(Muito\_fraco) \cdot P(Sab|Muito\_fraco)$$

$$P(Sab) = 0{,}0900 \cdot 0{,}1000 + 0{,}5300 \cdot 0{,}0900 + 0{,}3700 \cdot 0{,}2000 + 0{,}0100 \cdot 0{,}3000 = 0{,}1337$$

De forma análoga foram efetuados os cálculos para os demais dias da semana, e para as demais evidências, resultando nas suas respectivas probabilidades *a priori*, e consequentemente na rede bayesiana *a priori* apresentada na figura 5.4.

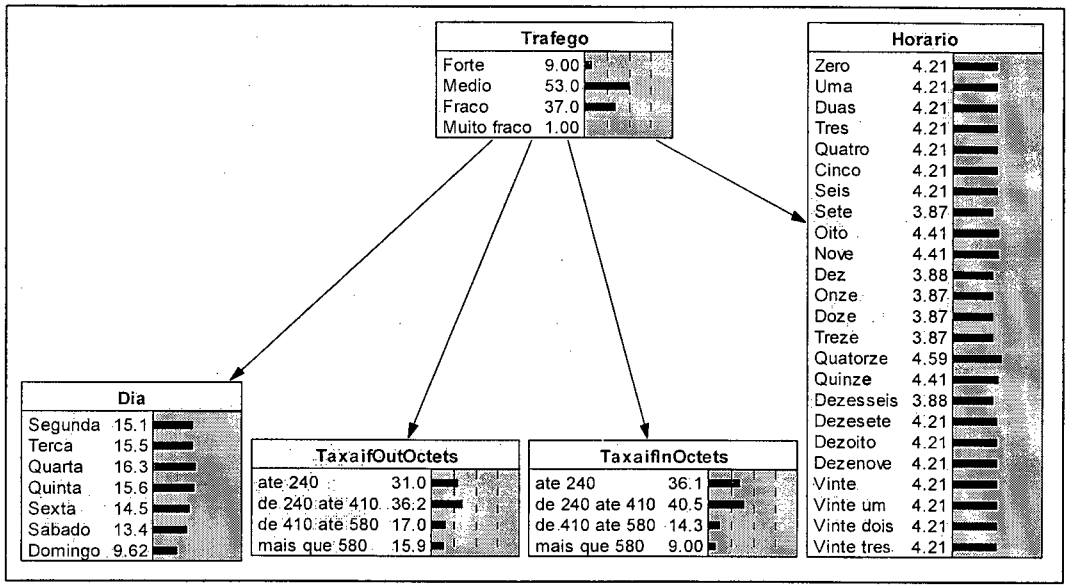

Figura 5.4: Rede Bayesiana *a priori* do sistema

5.7.2. Atualização da Rede Bayesiana para uma Nova Evidência

Dada a ocorrência de uma evidência a rede deve atualizar as probabilidades. Por exemplo, se for constatado que o dia da semana é segunda-feira. As probabilidades ficam conforme a tabela 5.8. Veja a demostração abaixo.

A probabilidade do tráfego ser Intenso dado que é segunda-feira é dada por:

$$P(Intenso|Seg) = \frac{P(Intenso) \cdot P(Seg|Intenso)}{P(Seg)} = \frac{0,0900 \cdot 0,2000}{0,1509} = \frac{0,0180}{0,1509} = 0,1193$$

A probabilidade do tráfego ser Normal se é segunda-feira é dada por:

$$P(Normal|Seg) = \frac{P(Normal) \cdot P(Seg|Normal)}{P(Seg)} = \frac{0,5300 \cdot 0,1800}{0,1509} = \frac{0,0954}{0,1509} = 0,6322$$

A probabilidade do tráfego ser Fraco se é segunda-feira é dada por:

$$P(Fraco|Seg) = \frac{P(Fraco) \cdot P(Seg|Fraco)}{P(Seg)} = \frac{0,3700 \cdot 0,1000}{0,1509} = \frac{0,0370}{0,1509} = 0,2452$$

A probabilidade do tráfego ser Muito_fraco se é segunda-feira é:

$$P(Muito\_fraco|Seg) = \frac{P(Muito\_fraco) \cdot P(Seg|Muito\_fraco)}{P(Seg)} = \frac{0,0100 \cdot 0,0500}{0,1509} = \frac{0,0005}{0,1509} = 0,0033$$

Tabela 5.8: Probabilidades das hipóteses diagnósticas *a posteori*

Hipóteses Diagnósticas	**P(Hi)**
Intenso	0,1193
Normal	0,6322
Fraco	0,2452
Muito_fraco	0,0033

Da mesma forma, as probabilidades das outras evidências também se alteram dada a certeza de ocorrência de uma das evidências.

A probabilidade da Taxa Média ifOutOctets ser menos do que 240 Kbps dado que é segunda-feira:

$P(TaxaifOutOctets\_até\_240) = P(Intenso_{/Seg} \cap TaxaifOutOctets\_até\_240) + P(Normal_{/Seg} \cap TaxaifOutOctets\_até\_240) + P(Fraco_{/Seg} \cap TaxaifOutOctets\_até\_240) + P(Muito\_fraco_{/Seg} \cap TaxaifOutOctets\_até\_240) = P(Intenso_{/Seg}) \cdot P(TaxaifOutOctets\_até\_240 | Intenso_{/Seg}) + P(Normal_{/Seg}) \cdot P(TaxaifOutOctets\_até\_240 | Normal_{/Seg}) + P(Fraco_{/Seg}) \cdot P(TaxaifOutOctets\_até\_240 | Fraco_{/Seg}) + P(Muito\_fraco_{/Seg}) \cdot P(TaxaifOutOctets\_até\_240 | Muito\_fraco_{/Seg}) = P(TaxaifOutOctets\_até\_240) = 0{,}0000 \cdot 0{,}1193 + 0{,}0000 \cdot 0{,}6322 + 0{,}8100 \cdot 0{,}2452 + 1{,}0000 \cdot 0{,}0033 = 0{,}2019$

De forma análoga são calculadas as probabilidades *a posteori* para todas as outras evidências, resultando a rede *a posteori* apresentada na figura 5.5.

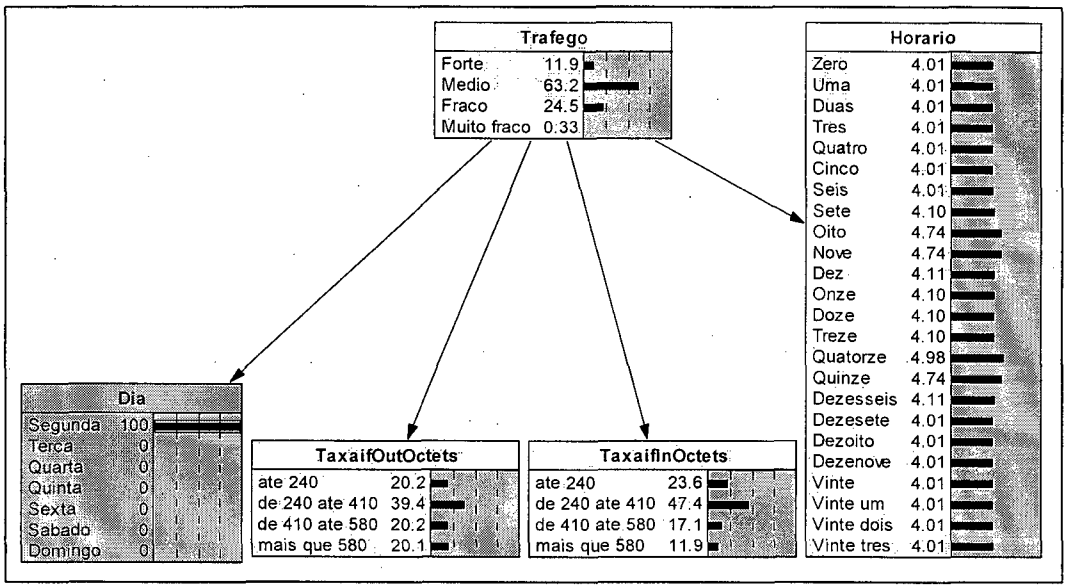

Figura 5.5: Rede Bayesiana *a posteori*, dado que é segunda-feira

Se além do evento Segunda-feira acrescentarmos que são oito horas as probabilidades seriam:

$$P(Intenso_{/Seg} | Oito\_horas) = \frac{P(Intenso_{/Seg} \cap Oito\_horas)}{P(Oito\_horas)} =$$

$$= \frac{P(Intenso_{/Seg}) \cdot P(Seg | Intenso_{/Oito\_horas})}{P(Oito\_horas)} = \frac{0{,}1193 \cdot 0{,}074}{0{,}0474} = 0{,}1762$$

$$P(Normal_{/Seg} | Oito\_horas) = (0,6322 \bullet 0,0500) / \ 0,0474 = 0,6669$$

$$P(Fraco_{/Seg} | Oito\_horas) = (0,2452 \bullet 0,0300) / 0,0474 = 0,1552$$

$$P(Muito\_fraco_{/Seg} | Oito\_horas) = (0,0033 \bullet 0,0200) / 0,0474 = 0,0014$$

Onde a probabilidade de haver tráfego às oito horas sabendo que é segunda-feira é dada por:

$P(Oito\_horas) = P(Intenso_{/Seg} \cap Oito\_horas) + P(Normal_{/Seg} \cap Oito\_horas) + P(Fraco_{/Seg} \cap Oito\_horas) + P(Muito\_fraco_{/Seg} \cap Oito\_horas) = 0,0700 \bullet 0,1193 + 0,0500 \bullet 0,6322 + 0,0300 \bullet 0,2452 + 0,0200 \bullet 0,0033 = 0,0474$

De forma análoga são calculadas as probabilidades *a posteori* para todas as outras evidências, resultando a rede *a posteori* apresentada na figura 5.6.

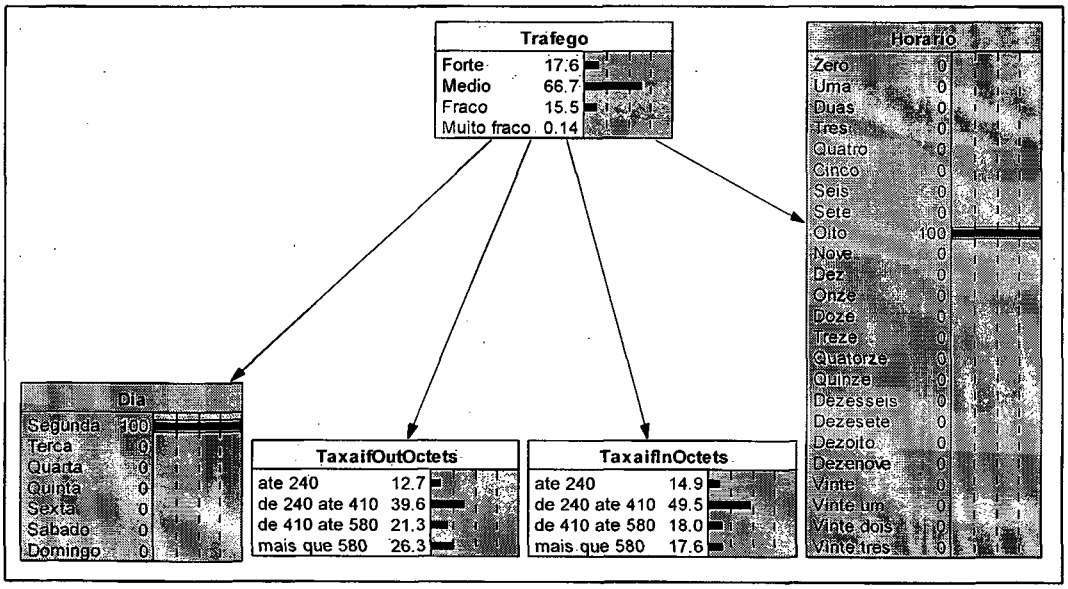

Figura 5.6: Rede Bayesiana *a posteori*, dado que é segunda-feira e oito horas

Veja a tabela 5.9, apresentado a evolução das probabilidades das hipóteses diagnósticas dadas as evidências.

Tabela 5.9: Evolução das Probabilidades das Hipóteses Diagnósticas

Hi	**P(Hi)**	**P(Hi/Seg)**	**P(Hi/ Seg e Oito_horas)**
Forte	0,0900	0,1193	0,1762
Média	0,5300	0,6322	0,6669
Fraca	0,3700	0,2452	0,1552
Muito_fraco	0,0100	0,0033	0,0014

Da forma como foi demonstrada, utilizando a rede bayesiana, podemos obter facilmente qualquer probabilidade sobre o tráfego da rede onde as probabilidades *a priori* vão se alterando através da aquisição de informação das evidências.

CAPÍTULO VI

6. Conclusões

6.1. Conclusões

Com os experimentos realizados e através do protótipo implementado constatou-se a adequação do enfoque probabilístico no desenvolvimento de um sistema especialista de apoio à gerência de redes. O protótipo implementado ajuda também a compreender melhor o raciocínio sob incerteza, podendo ser usado para o treinamento de futuros administradores.

Em acréscimo, o presente trabalho apresentou um novo conceito na área de Gerência de Redes, o conceito de "*Baselines* Dinâmicas". Onde a rede bayesiana implementada é utilizada para expressar o comportamento da rede, atualizando-se com as mudanças na mesma. Uma das vantagens da utilização da *baseline* implementada é que ela reflete o comportamento da rede através de probabilidades, ou seja, um determinado comportamento pode ser estimado a aprobabilidade de sua ocorrência e verificar se está dentro do esperado e não, como ocorre nas *baselines* convencionais, simplesmente estar ou não de acordo com o perfil da rede monitorada.

Relacionando o SISGEBAY com os trabalhos correlatos apresentados no capítulo 2, podemos observar que a *baseline* aqui implementada poderia ser utilizada pelos

sistemas especialistas implementados por [FRANCESCHI 97] e [ROCHA 97] adicionando as vantagens acima.

O trabalho apresentado em [KOEHLER 98] também utiliza o raciocínio bayesiano, só que com domínio de aplicação diferentes do domímio de aplicação deste trabalho. No trabalho de [KOEHLER 98] o domínio de aplicação é a área médica e uma das vantagens de seu trabalho é que o sistema implementado é capaz de realizar o diagnóstico considerando poucas entradas. Esta vantagem também pode ser observada no SISGEBAY.

Considerando o processo de gerência pró-ativa, o SISGEBAY apresenta uma grande contribuição. Como ele pode realizar diagnóstico considerando poucas entradas e é capaz de predizer o comportamento da rede, os problemas de tráfego podem ser identificados antes de sua ocorrência e as ações necessárias para deter ou amenizar estes problema podem ser tomadas antes de sua ocorrência – o que é a base do conceito de gerência pró-ativa.

6.2. Trabalhos Futuros

Com base nos resultados obtidos neste trabalho pode-se recomendar uma série de trabalhos futuros. Alguns possíveis trabalhos foram melhor estudados e estão detalhados abaixo, dentre eles destaca-se a aplicação de *baselines* bayesianas em gerência de redes distribuída.

6.2.1. Ampliação das Variáveis e Segmentos Monitorados

Pode-se ampliar o número de variáveis gerenciais monitoradas, e assim predizer o comportamento do segmento de rede monitorado, não somente quanto ao tráfego, mas também quanto a outros ítens de gerenciamento.

Aumentando o número de segmentos monitorados, pode-se predizer o comportamento da rede como um todo ou em segmentos individuais, ficando assim mais fácil detectar futuras falhas.

O problema de monitorar um número muito grande de variáveis e segmentos é que o tráfego da rede é alterado (aumentado) devido às coletas. Uma solução possível para este problema pode ser a distribuição da gerência.

6.2.2. Distribuição da Gerência

A coleta de dados geralmente sobrecarrega e aumenta o tráfego da rede.

Durante este trabalho foi estudada a possibilidade da distribuição da gerência utilizando a arquitetura OMG CORBA para distribuir o gerenciamento[LORENSET 98]. Obteve-se um modelo de gerenciamento que pode ser implementado como trabalho futuro utilizando a *baseline* bayesiana.

Na arquitetura descrita a seguir o suporte ORB CORBA é o meio que possibilita o relacionamento gerente-agente tal qual é conhecido no modelo de gerenciamento clássico.

Usando um ORB, a *Management Unit* (MU) pode transparentemente invocar *o* agente, que pode estar em uma *Functional Management Unit* (FMU), sobre a Internet, como pode ser visto na figura 6.1.

É importante notar que as regras gerente-agente são utilizadas para coordenar as interações de gerenciamento da rede.

6.2.2.1. Arquitetura do Sistema de Gerência Distribuída

A figura 6.1 mostra a arquitetura do sistema de gerência distribuída proposto para trabalhos futuros. ME significa *Managed Equipament* e FMU significa *Functional Management Unit.*

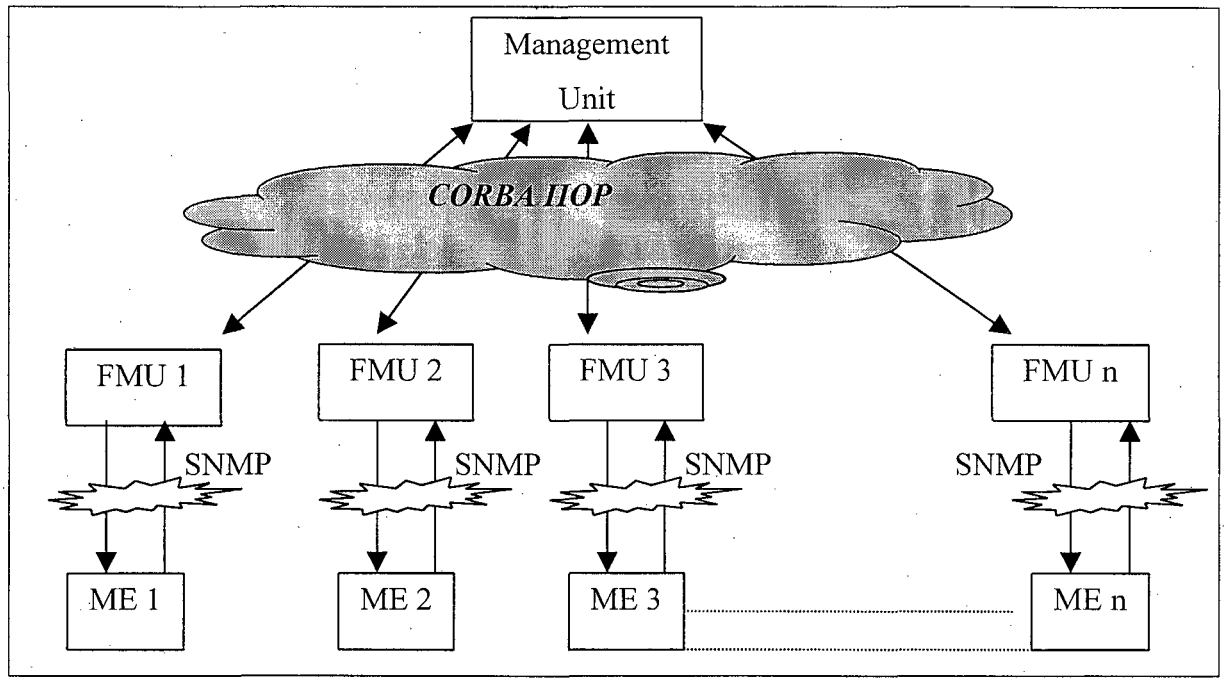

Figura 6.1: Arquitetura do sistema de gerência distribuída

A *Management Unit* (MU) gerencia as *queries* do sistema.

Dada uma das FMUs têm a responsabilidade de gerenciar uma ME. Você pode ter mais que uma FMU em um mesmo *host*, gerenciando desta forma mais que uma ME por *host*. Cada FMU é responsável por enviar e processar as informações necessárias para a MU.

CAPÍTULO VII

7. Referências Bibliográficas

[BARRETO 96] BARRETO, J.M. "Inteligência Artificial – No Limiar do Século XXI". Florianópolis – SC. 1997.

[BERNHARDT 96] BERNHARDT, M. "Design and implementation of a web-based tool for ATM connection management". Stuttgart, Aug. 1996. Master's Thesis – Departament of Computer Science, University of Stuttgart.

[BOWERMAN 87] BOWERMAN, B.L.; OÇONNEL, R.T. "Time Series Forecasting – Unified Concepts and Computer Implementation". PWS Publishers, 1987.

[CARVALHO 93] CARVALHO, Tereza Cristina Melo de Brito et alli – Gerenciamento de Redes: Uma Abordagem de Sistemas Abertos. BRISA (Sociedade Brasileira para Interconexão de Sistemas Abertos), Makron Books, 1993.

[CHATFIELD 84] CHATFIELD, C. "The Analysis of Time Series". Chapman and Hall, 1984.

[CHEESEMAN 85] CHEESEMAN, P. "In defense of probability". Proceedings of 9th International Joint Conference on Artificial Intelligence.

Los Angeles, pp 1002-1009. 1985.

[CISCO 99] CISCO. URL: http://cisco.com/warp/public/733/7000/, janeiro de 1999.

[CLOCKSIN 84] CLOCKSIN, W.F.; MELLISH, C.S. "Programing in Prolog". Second edition. Cambridge, England, Springer – Verlag, 1984.

[COHEN 85] COHEN, P. R. "Heuristic reasoning about uncertainty: artificial intelligence approach". Boston: Pitman. 1985.

[DACONTA 96] DACONTA, M. "Java for C/C++ programmers". USA: Ellist, 1996.

[FAYYAD 96] FAYYAD, U.M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. "Advances in Knowledge Discovery and Data Mining". American Association for Artificial Intelligence, Menlo Park, California EUA. 1996.

[FRANCESCHI 97] FRANCESCHI, [a]S.M.; ROCHA, M.[a];WEBER, H.L.; WESTPHALL, C.B. "Employing Remote Monitoring and Artificial Intelligence Techniques to Develop the Proactive Network Management". Proceedings of the International Workchop on Applications of Neural Networks to Telecommunication 3. Laurence Erlbaum Associates, Publishers. Mahwah, (NJ), USA. 1997.

[FRAWLEY 91] FRAWLEY, W.J.; PIATETSKY-SHAPIRO, G.; AND MATTHEUS, C.J. 1991. "Knowledge Discovery in Datavases: Na Overview". In Knowledge Discovery in Databases, ed. G. Piatetsky-Shapiro and B. Frawley. Cambridge, Mass: AAAI/MIT Press, 1-27.

[HECKERMAN 95] HECKERMAN, D. "A Bayesian Approach to Learning Causal".

Technical Report. MSR-TR-95-04, Microsoft Research, March, 1995.

[HP 00] HP – Hewlett Packard. URL: http://www.openview.hp.com/index.asp, janeiro de 1999 / fevereiro de 2000.

[IETF 99] IETF – Internet Engening Task Force URL: http://www.ietf.cnri.reeston.va.us/rfc/rfc1213.txt, janeiro de 1999.

[KNOBBE 97] KNOBBE, A. J. "Data Mining for Adpative System Management". In proceedings of PADD. 1997.

[KOEHLER 98] KOEHLER, C. "Uma abordagem probabilística para sistemas especialistas". Dissertação de Mestrado. UFSC. Florianópolis – SC. 1998.

[LANGLEY 95] LANGLEY, P. & SIMON, H.A. "Applications of machine learning and rule induction". Communications of the ACM. Vol. 38, No 11. November 1995.

[LAURITZEN 88] LAURITZEN, S. L. & SPIEGELHALTER, D. J. "Local computations with probabilities on graphical structures and their applications to expert systems". J. Royal Statist. Soc., B, 50(2):154-227. 1988.

[LINDA 96] LINDA, C. Van Der Gaag. "Bayesian Belief Networks: Odds and Ends". The Computer Journal. Vol. 39. No 02. 1996.

[LINDLEY 82] LINDLEY, D. V. "Scoring rules and the inevitability of probability". International Statistical Review, (50):1-26. 1982.

[LORENSET 98] LORENSET, V.L. "TMN Distributed Management: an experience in alarm furveillance with CORBA". Master

Thesis, CPGCC/UFSC, 1998.

[MEGAPUTER 00] MEGAPUTER. URL: http://www.megaputer.com , setembro de 1998 / fevereiro de 2000.

[MICROSOFT 99] MICROSOFT. URL: http://microsoft.com/office/, janeiro de 1999.

[NASSAR 98] NASSAR, S.M. "Estatística e Informática: Um Processo Iterativo Entre Duas Ciências". Trabalho apresentado no concurso para professor titular. Departamento de Informática e Estatística. Centro de Tecnologia. Universidade Federal de Santa Catarina. 128p. Abril, 1998.

[NETICA 00] NETICA. URL: http://www.norsys.com, janeiro de 1999 / fevereiro de 2000.

[NETO 98] NETO, F.W. "Aplicando a Técnica de Séries Temporais em Gerenciamento Pró-Ativo de Redes de Computadores". Anais do Simpósio Brasileiro de Redes de Computadores. Rio de Janeiro (RJ). Maio de 1998.

[PEARL 88] PEARL, J. "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference". San Mateo, Calif.: Morgan Kaufmann, 1988.

[RCT 00] RCT – Rede de Ciência e Tecnologia de Santa Catarina. URL: http://www.pop-ufsc.rct-sc.br, janeiro de 1999 / fevereiro de 2000.

[RNP 00] RNP - Rede Nacional de Pesquisa. URL: http://www.rnp.br, janeiro de 1999 / fevereiro de 2000.

[ROCHA 97] ROCHA, M.A.; WESTPHALL, C.B. "Proactive Management of Computer Networks using Artificial Intelligence Agents and Techniques". Proceedings of the Symposium on

Integrated Network Management. San Diego (CA), USA. May, 1997.

[SÁENS 96] SÁENS, A.E. "Sistema Especialista para a Gerência Pró-Ativa de Redes". Tese de Mestrado, UFRGS, 1996.

[SAMPAIO 97] SAMPAIO, S.C. "Plataforma para concepção de aplicações de gerência utilizando o SNMP". Projeto Específico. UNIFACS – Universidade Salvador S/C. Salvador – BA, 1997.

[SHAFER 76] SHAFER, G. "A mathematical theory of evidence". Princeton, Princeton University Press. 1976.

[SUN 00] SUN - URL: http://www.sun.com, janeiro de 1999 / fevereiro de 2000.

[TIVOLI 99] TIVOLI NetView. URL: http://www.tivoli.com/o_products/html/netview.html, janeiro de 1999.

[VERONEZ 99] VERONEZ, C.A.; EFRAIN, C.; BAROTTO, A.M.; NASSAR, S.M.;WESTPHALL, C.B. "Gerência de Redes Utilizando Métodos Estatísticos Bayesianos". Anais do Simpósio Brasileiro de Redes de Computadores. Salvador (BA), Brasil. Maio de 1999.

[WESTPHALL 96] WESTPHALL, C.B.; KORMANN, L.F. "Usage of the TMN Comcepts for Configuration Management of ATM Network". International Symposium on Advanced Imaging and NetWork Technologies. Berlim, Alemanha Out. 7-11, 1996.

[WESTPHALL 91] WESTPHALL, C.B. "Conception et développement de l'architecture d'administration d'un réseau métropolitain". Thèse de doctorat nouveau régime. L'université Paul

Sabatier. Toulouse, le 16 juillet 1991.

[WESTPHAL 98] WESTPHAL, C., BLAXTON, T. "Data Mining Solutions – Methods and Tools for Solving Real-World Problems". John Wiley & Sons, Inc. New York, N.Y. USA. 1998.

[ZADEH 83] ZADEH, L. A. "The role of fuzzy logic in the management of uncertainty in expert systems". Fuzzy Sets and Systems. (11):199-228. 1983.